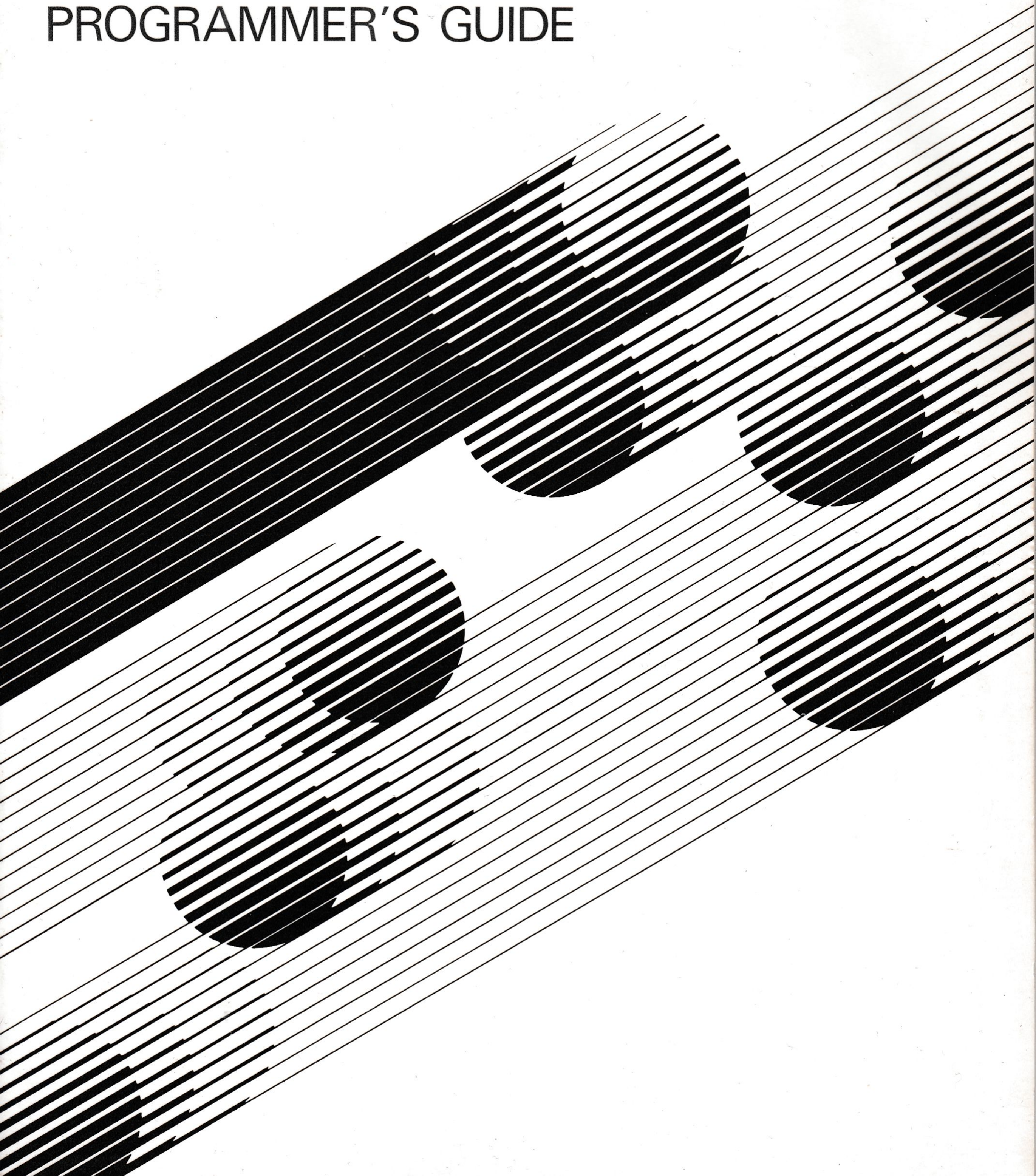
NEC Personal Computer

PC-8801FH/MH

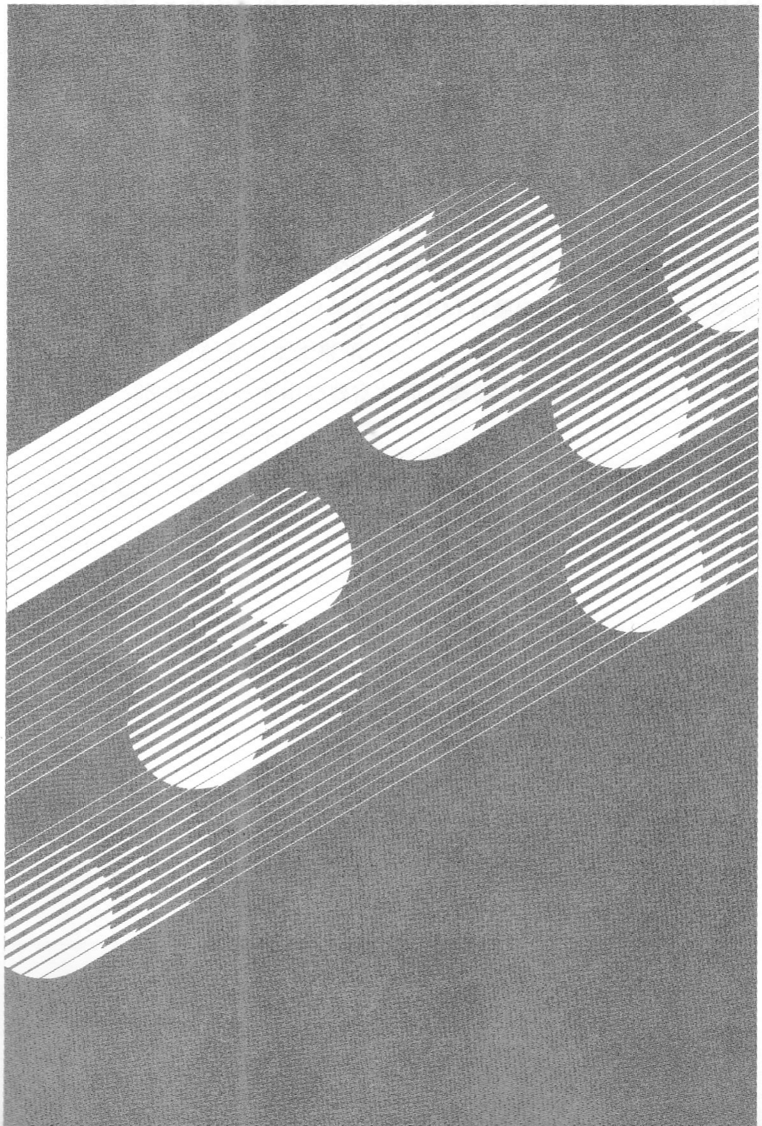
PROGRAMMER'S GUIDE



PC-8801 FH/MH-PG

PC-8801FH/MH

PROGRAMMER'S GUIDE



はじめに

本書は，N_{88}-BASICシステムとN_{88}-日本語BASICシステムでプログラムを作る人のために，本機が提供しているプログラミングの環境と本機を使いこなすうえでの初歩的な手順から高度・実用的なプログラムまで幅広く説明しています．

本機の特長を下記に示します．

- ●8MHzのクロックで動作する8bit CPU μPD70008C-8の搭載により高速処理を実現するとともに，従来機種との互換性を保つための4MHzクロック動作モードをサポート．
- ●利用者サイドの設計思想が反映され，いろいろなバリエーションが選べる豊富な機種(メモリ容量によるバリエーション，供給メディアによるバリエーション)．
- ●JIS第2水準漢字ROM標準実装，日本語入力対応キーボード採用(シリーズ共通)，文節変換(PC-8801MH)，熟語変換(PC-8801FH)が可能なN_{88}-日本語BASIC標準添付による日本語への本格対応(ただし，PC-8801FHのモデル10は除く)．
- ●PC-8801 mkⅡFR/PC-8801mkⅡMRのすぐれた機能(アナログ対応の高速グラフィックス機能，FM音源によるデジタルシンセサイザ機能)をそのまま継承．
- ●PC-8800シリーズ用の豊富な周辺機器とアプリケーションソフトウェアをサポート．

本書が実際のプログラミングのうえでお役に立てば幸いです．

本書の見方

本書は，N₈₈-BASICシステムおよびN₈₈-日本語BASICシステムの機能について説明しています．各章の主な内容は下記のとおりです．

第1章，第2章では，N₈₈-BASICシステムおよびN₈₈-日本語BASICシステムの概要と起動方法について記述しています．

第3章，第4章では，本機のキーボードの機能と，N₈₈-BASICシステムおよびN₈₈-日本語BASICシステムでのスクリーンエディタについて記述しています．

第5章，第6章，第7章では，N₈₈-日本語BASICシステムでの日本語入力の概要と操作法および日本語ユーティリティについて記述しています．

第8章，第9章，第10章，第11章では，N₈₈-BASICシステムおよびN₈₈-日本語BASICシステムでのテキスト画面に対する文字の表示とグラフィックス機能，カラー表示，カラーコピーユーティリティについて記述しています．

第12章では，N₈₈-BASICシステムおよびN₈₈-日本語BASICシステムでのサウンド機能について記述しています．

第13章，第14章，第15章では，N₈₈-BASICシステムおよびN₈₈-日本語BASICシステムでのファイルについての概念と操作法について記述しています．

第16章では，本機で使用できるディスク関係のユーティリティプログラムの操作方法について記述しています．

第17章では，本機の通信モードの概要と操作法について記述しています．

第18章では，N₈₈-BASICシステムおよびN₈₈-日本語BASICシステムでのBASICプログラムのデバッグの手法について記述しています．

第19章では，N₈₈-BASICシステムおよびN₈₈-日本語BASICシステムでの機械語モニタの操作法について記述しています．

第20章では，N₈₈-BASICシステムおよびN₈₈-日本語BASICシステムでの機械語プログラムの作り方について記述しています．

資料編では，キャラクタコード表やコントロールコード表など，N₈₈-BASICおよびN₈₈-日本語BASICを使いこなすうえで，有用な資料となるものをまとめて記載しています．

このマニュアルで使用している用語

本機は強力な2つのBASIC—N_{88}-BASICとN_{88}-日本語BASIC—を装備しています.

N_{88}-BASICはいくつかのモードを持ち，また，ディスク装置を使う場合，使わない場合などでも少しずつ機能が異なります.

ここでは，BASICのバージョンやモードを中心として，このマニュアルに使用している用語をまとめてみました.

1. BASICの種類やモード名

用　語	意　味
N_{88}-BASIC V1	従来のN_{88}-BASIC（PC-8801，PC-8801MKII 用のN_{88}-BASIC）と互換性があるモード．本機のシステムモードスイッチをV1SまたはV1Hにセットしてスタートさせる． N_{88}-BASICシステムディスクを使ってスタートさせるV1 DISKと，ディスクを使わないV1 ROMの2つのモードの総称.
N_{88}-BASIC V2	PC-8801MKIISR/FR/MR用の，N_{88}-BASICの互換性のあるモード．本機のシステムモードスイッチをV2にセットしてスタートさせる． N_{88}-BASICシステムディスクを使ってスタートさせるV2 DISKと，ディスクを使わないV2 ROMの2つのモードの総称.
N_{88}-日本語BASIC	N_{88}-BASIC V2に，日本語処理機能を付加して，従来のPC-8801MKIIMR/FRと互換性のあるBASIC.本機のシステムモードスイッチをV2にセットし，N_{88}-日本語BASICシステムディスクを使ってスタートさせる．ディスクドライブに，システムディスクをセットした状態で動作する.
N_{88}-BASIC DISK version	V1 DISKとV2 DISKの総称.
N_{88}-BASIC ROM version	V1 ROMとV2 ROMの総称.
BASIC	パソコンで一般的に広く使われているBASICすべての名称. 狭義には本機のN_{88}-BASIC V1，V2, N_{88}-日本語BASICを意味する.
拡張命令	N_{88}-BASIC V2およびN_{88}-日本語BASICで，NEW CMDコマンドを実行することにより，使用可能になるサウンド機能やアナログパレット機能をサポートする命令群.

2. N_{88}-日本語BASICで使われる用語

用　　　　語	意　　　　　　　味
日本語モード	SCREEN 文の第1パラメータを3，または4に指定したときのモード．全角文字の画面表示やキー入力が可能． N_{88}-日本語BASICのスタート時は，標準ディスプレイ使用のときSCREEN 3，専用ディスプレイ使用のときはSCREEN 4に設定されている．
グラフィックシンボルモード	日本語モードに対するモードで，SCREEN 文の第1パラメータを，0，1，2に指定したときのモード．全角文字の画面表示，キー入力はできない．全角文字に相当するコードを表示させると，グラフィックシンボルを含む半角文字が表示される．
全角文字	日本語を表すための文字（日本語コードにあげられている文字）．BASICはシフト JIS コードによって全角文字を扱い，MID$ などの関数は，全角文字1文字を半角文字2文字と等価に扱う．
半角文字	英数字，カタカナ，コントロールコードを含む文字（キャラクタコードにあげられている256文字）． N_{88}-BASICなどの日本語処理機能を持たないBASICでは，半角文字しか使用できない．
全角文字入力モード 半角文字入力モード	日本語モードでは，全角文字と半角文字とを混在して使用することができる．キー入力時には 全角 によって，全角文字を入力するためのモード（全角文字入力モード）と，半角文字を入力するモード（半角文字入力モード）とを切り替えながら行う．

目　次

第15章　フロッピーディスクとファイルの構造

第16章　ディスク関係のユーティリティプログラム

第17章　コンピュータコミュニケーション

第18章　デバッグ

第19章　機械語モニタ

第20章　機械語プログラムを呼ぶ

資　料

ユーティリティ目次

この目次は、システムディスクに入っているユーティリティを使う便宜のために、各ユーティリティの操作法が、『ユーザーズガイド』『プログラマーズガイド』のどこに掲載されているかを示したものです。

UG　ユーザーズガイドの略
PG　プログラマーズガイドの略

(1) ディスクユーティリティ

フロッピーディスクのコピー	（その１） (copy1 . j88)[注] copy2 . j88	UG 3 章 5 節
	（その２） dskut2 . j88 (dskut1 . j88)[注]	PG 16章 7 節
システムディスクの作成	dskut2 . j88 (dskut1 . j88)[注]	PG16章 7 節
データディスクの作成	dskut2 . j88 (dskut1 . j88)[注]	PG16章 7 節
IDセクタの書き換え	dskut2 . j88 (dskut1 . j88)[注]	PG16章 7 節
ファイル転送	xfiles . j88	PG16章 7 節
ディスクコードのコピー	sysgn2 . j88 (sysgn1 . j88)[注]	PG16章 7 節

(2) N88-日本語BASICユーティリティ

外字管理	font . mh (fh)[注]	PG 7 章 2 節
プリンタへの外字ロード	prload . mh (fh)[注]	PG 7 章 2 節
ユーザ辞書管理	usrdic . mh (fh)[注]	PG 7 章 4 節
コード変換	cdconv . mh (fh)[注]	PG 7 章 5 節

(3) サウンドユーティリティ

スーパーキーボードシステム	sks	PG12章 4 節
ミュージックエディタキット	mek	PG12章 4 節

(4) カラーコピーユーティリティ

カラーコピーユーティリティ	@cc2 . j88	PG11章
	@cc4v . j88	
	@cc4h . j88	
	@cc4m . j88	

注 ファイル名の( )はPC-8801FHの場合のみを示します.

第1章

N_{88}-BASIC
N_{88}-日本語BASIC
の世界と特徴

本機は，標準プログラミング言語として，N_{88}-BASICとN_{88}-日本語BASICの2種の言語を備えています．
N_{88}-BASICには，V1（バージョン1）とV2（バージョン2）の2つのモードがあります．V2モードは，V1モードの諸機能に加え，グラフィック機能，サウンド機能，アナログパレット機能などを強化したものです．N_{88}-日本語BASICは，N_{88}-BASIC（V2）の機能をさらに強化したもので，日本語の処理が行えます．
本章では，以上の種々のBASICの関係と，それぞれのBASICの特徴の概略をご紹介します．

1 本機で使用できる各BASICの関係

1. 本機で使用できる各BASIC

本機では，次の３つのBASICを使用することができます．

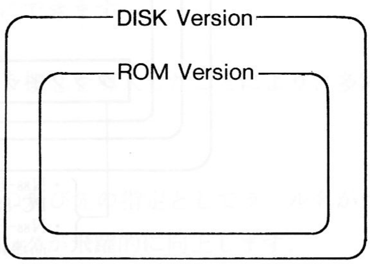
(i) N_{88}-BASIC V1

(ii) N_{88}-BASIC V2

(iii) N_{88}-日本語BASIC

このうち，N_{88}-BASIC V1とN_{88}-BASIC V2には，それぞれROM VersionとDISK Versionがあります．N_{88}-日本語BASICはDISK Versionのみで，ROM Versionはありません．

ROM Versionとは本体内蔵のROMだけで作動するもののことで，DISK VersionとはN_{88}-BASICシステムディスクを使い，ディスクから読み出した情報とROMの情報の双方をもとに作動するもののことです．V1, V2モードとも，DISK VersionはROM Versionの全機能を含んでいます．

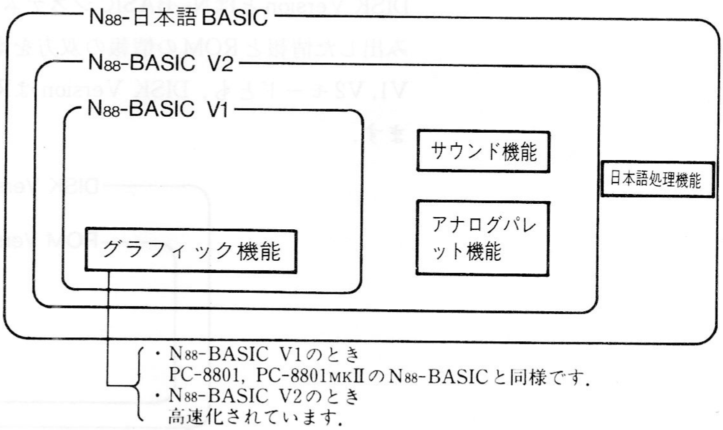


2. 各BASICの関係

各BASICの関係をまとめると，次の表のようになります。

BASIC名	Version	略　称	媒　体	モードスイッチ	互換性
N_{88}-BASIC V1	N_{88}-BASIC V1 ROM Version	V1 ROM	ROM	V1 (V1H,V1S)	PC-8801 PC-8801mkII
	N_{88}-BASIC V1 DISK Version	V1 DISK	ROM + N_{88}-BASIC システムディスク		
N_{88}-BASIC V2	N_{88}-BASIC V2 ROM Version	V2 ROM	ROM	V2	PC-8801 PC-8801mkII PC-8801mkIISR PC-8801mkIIFR PC-8801mkIIMR PC-8801mkIITR 本機
	N_{88}-BASIC V2 DISK Version	V2 DISK	ROM + N_{88}-BASIC システムディスク		
N_{88}-日本語BASIC	～	なし	ROM + N_{88}-日本語BASIC システムディスク		

また，機能面から見たBASICの関係を図で示すと，次のようになります．



2 N_{88}-BASIC V1の特徴

N_{88}-BASIC V1は従来のN_{88}-BASICと互換性がありますので，PC-8801用，PC-8801mkⅡ用のソフトウェアのほとんどを使用することができます．

1．グラフィックス機能の強化・高速化

(1) 48Kバイトものグラフィックス専用RAMを持っているため，640×200ドット（8色カラー1ページまたは白黒3ページ），640×400ドット（白黒1ページ）の高分解能グラフィックスを実現できます．

(2) またこれをサポートする種々のグラフィック命令は強力で，特にWINDOW, VIEWによる論理的な座標系の設定は，プログラミングを簡略にし，表示を多彩にします．

(3) リストなどキャラクタを表示するテキストRAMと，グラフィックRAMは完全に区別されているため，テキスト画面とグラフィック画面の合成が簡単にできます．

2．パレット機能

(1) パレットという概念を導入したことにより，多彩なカラー表示が可能です．

3．その他の機能

(1) プログラム中の飛び先の指定としてラベル名が使えるため，プログラム開発，デバッグが飛躍的に向上します．

(2) IF THEN ELSE/WHILE WEND等の文により構造化されたプログラムにすることができます．

(3) 倍精度関数が準備されており，科学技術計算にも適応できます．また，大規模なプログラムに対してオーバーレイ機能も用意されており，必要な変数の受け渡しも可能です．

(4) 各ハードウェアからの割り込みをBASIC中で操作できます．

(5) 本機が持っているI/OインタフェースをN_{88}-BASICで制御できます．操作可能なI/Oインタフェースは次のとおりです．

(a) RS-232Cインタフェース

(b) ミニフロッピーディスク

(c) 8インチフロッピーディスク

(d) セントロニクス・インタフェース（プリンタ）

(e) CMTインタフェース（PC-8801FHの場合）

3 N_{88}-BASIC V2の特徴

N_{88}-BASIC V2は，本機のハードウェアに対応するために，N_{88}-BASIC V1を機能拡張したものです．

N_{88}-BASIC V1と上位互換性があり，N_{88}-BASIC V1で書かれたソフトウェア(BASICで書かれたもの)を使用することができます．また，PC-8801mkⅡSR用，PC-8801mkⅡFR用，PC-8801mkⅡMR用のソフトウェアも使用することができます．

機能拡張された点は，次のとおりです．

1．グラフィックス機能の強化・高速化

超高速CPU(8MHz)の採用により，グラフィックス処理が高速化されています．

2．パレット機能

N_{88}-BASIC V2では，本体にアナログRGBディスプレイを接続することにより，同一画面に512色の中から8色を選んで表示することができます(アナログパレット機能)．

3．サウンド機能

シンセサイザIC(略称OPN……FM Operator Type-N)が装備されており，N_{88}-BASIC V2ではFM音源とSSG音源による6重奏が可能です．

また，OPNを直接制御することによって効果音を出力することができます．さらに，オーディオアンプなどを接続することによって，より本格的な演奏を楽しむことができます．

4．日本語表示機能

PUT@KANJI文が用意されており，JIS第1水準までの漢字を表示することができます(N_{88}-日本語BASIC使用時は，JIS第2水準まで表示できます．詳細は次頁参照)．

5．その他の機能

変数名は40文字まで使用可能であるため，わかりやすい変数名を付けることができます．

4 N88-日本語BASICの特徴

N88-日本語BASICは，N88-BASIC V2の機能をさらに拡張したものです．ハード的にも，本機においては日本語入力に適したキーボードの採用，JIS第２水準漢字ROMの標準装備により，日本語表現の可能性と操作性を高めています．

N88-BASIC V2と互換性があり，N88-BASIC V2で書かれたソフトウェア(BASICで書かれたもの)や，PC-8801mkⅡSR，PC-8801mkⅡFR，PC-8801mkⅡMR用のソフトウェアも使用することができます．また，PC-8801MHの日本語BASICはPC-8801FHの日本語BASICと互換性を持ち，PC-8801FHのN88-日本語BASICで書かれたソフトウェアを使用することができます．

N88-BASIC V2より機能拡張された点は以下のとおりです．

注意 N88-日本語BASICは，フロッピーディスクドライブがついていないと使用できません．

1. 全角文字の入力が可能

漢字やひらがな，カタカナなどの全角文字が使用できます．全角文字としては，JIS第１水準・JIS第２水準の漢字(6349語)と漢字以外の文字453種が使用できます．

2. 2通りの入力方式

全角文字の入力には，かな入力，ローマ字入力の２通りの方式が使用できます．

3. 熟語または文節単位で漢字変換

入力したかなは，文法処理を行いながら文節単位で(PC-8801MH)，熟語単位(PC-8801FH)で漢字に変換していくので，容易に日本語の文章を作成することができます．

4. 外字処理が可能

添付のN88-日本語BASICシステムディスクによって外字処理が可能です．

5. 独自のステートメント，関数の追加

KLEN(文字列に含まれる文字数を返す)など，N88-日本語BASIC独自の命令を追加しました．これにより，N88-日本語BASICの活用の範囲が広がります．

第2章

N₈₈-BASIC V2 の起動/終了とモード設定

本章ではN₈₈-BASICシステムディスクとN₈₈-日本語BASICシステムディスクを使用してN₈₈-BASICとN₈₈-日本語BASICの起動や終了の手順，本機をいろいろな目的で使用する上で知っておかなければならないスイッチの設定方法を説明します．

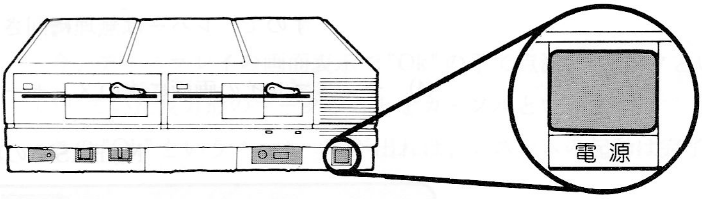
1 N$_{88}$-BASIC V2システムの起動

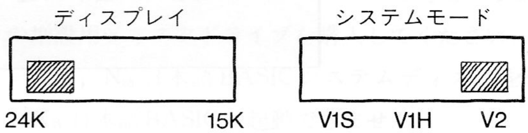
システムディスクを使って本機を起動させます．システムディスクには，N$_{88}$-BASICシステムディスクとN$_{88}$-日本語BASICシステムディスクの２種類があります．

起動時の初期画面に違いはありますが，起動方法はどちらも同じですから，起動の操作をする際にはどちらのシステムディスクを使用してもかまいません．

1．電源ON

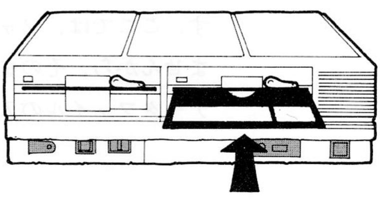
(1) ディスプレイスイッチなどのハードスイッチが下図に示すようになっているのを確認し，ディスプレイ，本体の順に電源ONにします．



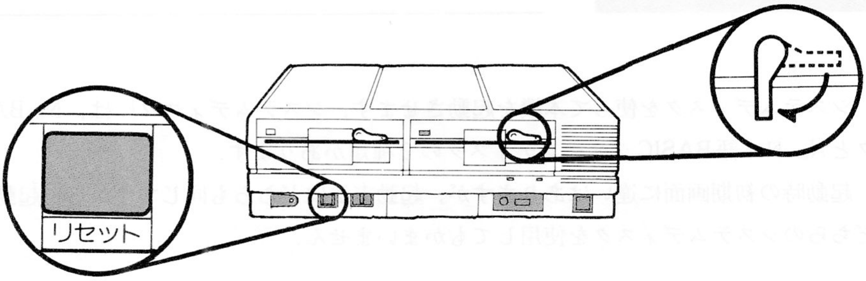


ディスプレイスイッチの設定は，「ユーザーズガイド　第１章」を参照してください．

(2) BASICモードランプ，ディスプレイの電源ランプがついているのを確認して，システムディスクをドライブ１に差し込みます．このとき，システムディスクをカチッと軽い音がするまで静かに差し込んでください．



(3) 本体のリセットボタンを押してから，ドライブランプが消えている間にドライブ1のレバーを下図に示すようにしめます．もし，レバーがしまらない場合は，もう1度システムディスクをきちんと奥まで差し込んでください．



注意 フロッピーディスクが入っていないときは，レバーがしまらないようになっています．無理に回すと故障の原因になりますので，レバーは無理に回さないでください．

(4) ディスプレイの画面に次のメッセージがあらわれます．

N88-BASICの場合

```
Disk version [Oct. 20,1986]
How many files(0-15)?█
```

N88-日本語BASICの場合

```
How many files(0-15)?█
```

"How many files (0-15)?"は，コンピュータがあなたに"同時にいくつのファイルを使いますか？"という意味のことを尋ねているのです．ここでは，ファイル数をいくつに設定してもプログラムに影響しませんから，そのまま ⏎ を押します(ファイルについては「第13章 データファイルの入出力」を参照してください)．

(5) ディスプレイの画面は次のようになります.

N88-BASICの場合

```
Disk version [Oct 20,1986]
How many files(0-15)?
NEC N-88 BASIC Version 2.3
Copyright (C) 1981 by Microsoft
***** Bytes free
Ok█
```

N88-日本語BASICの場合

```
N88-日本語ＢＡＳＩＣ＊＊ Version 2.0
Copyright (C) 1981 Microsoft Corporation
Copyright (C) 1985,1986 日本電気株式会社
使用可能メモリ容量：*****バイト
Ok█
```

今，ディスプレイの画面表示で"Ok"の下で点滅(ブリンクといいます)していた長方形のマークのことをカーソルといいます.

この"Ok"というメッセージが出れば，システムの起動は完了です.

注意 PC-8801FHのモデル10で，N88-BASICシステムディスクやN88-日本語BASICシステムディスクを使用する場合，別売りの増設用ディスクドライブを購入してください.

なお，N88-日本語BASICシステムディスクを使用しないとN88-日本語BASICは起動できません.

2 N88-BASICシステムの終了

自分のプログラムが完了したり，途中で一度中断して電源をOFFにするときは次のようにします．実際にプログラムを作る前に，電源OFFの手順をしっかり覚えておきましょう．

1. 電源OFFにする前に

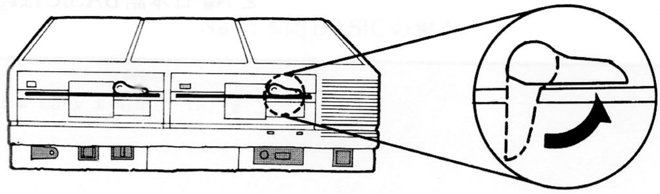
電源をOFFにする前に，自分で作った(あるいは，作りかけの)プログラムやデータをフロッピーディスクに記録(このことを書き込むといいます)することを忘れていないかを確認します(記録方法は「ユーザーズガイド　第4章11節」を参照)．

2. ファイルをCLOSEする

ファイルをOPENした場合は，ファイル操作終了後，必ずCLOSEを実行してください．特にディスクでは，ファイルをOPENしたまま(CLOSEせずに)ディスクを差しかえると，新たに挿入したディスクのデータをこわしてしまう危険性があります(CLOSE，OPEN命令の詳細は「リファレンスマニュアル」を参照)．

3. システムディスクを抜き取る

ドライブランプが消えていることを確認したら，レバーを動かしてシステムディスクを抜き出してください．



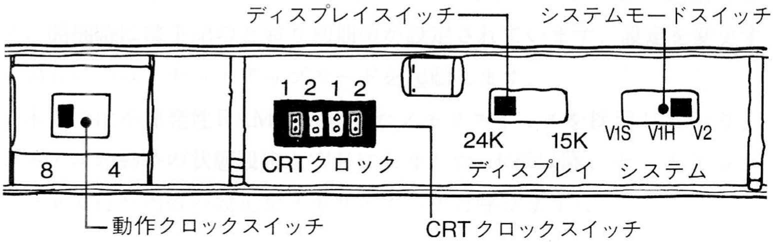
4. 電源OFF!!!

ディスプレイ，本体の順に電源をOFFにします．

3 ハードスイッチの設定

1. 4つのハードスイッチ

電源スイッチをONにする前に設定しなければならない4つのスイッチがあります.



(1) システムモードスイッチ

3ポジションのスライドスイッチで，システム起動時のN_{88}-BASICの動作モードを設定します.

システムモードスイッチの設定方法は「ユーザーズガイド　第4章」を参照してください.

(2) 動作クロックスイッチ

2ポジションのスライドスイッチで，メインCPUの動作クロックを選択します.

(3) ディスプレイスイッチ

2ポジションのスライドスイッチで，専用ディスプレイ／標準ディスプレイを選択します.

ディスプレイスイッチの設定については，「ユーザーズガイド　第1章」を参照してください.

(4) CRTクロックスイッチ

2つのジャンパスイッチで，1つはCRTクロックの内部/外部クロックを選択します．もう1つはデジタルRGBコネクタ内クロックの方向を選択します.

2. ハードスイッチの初期設定

ハードスイッチの開梱時の初期設定は，下図に示すアミの項目です．

ハードスイッチ	選択内容
システムモードスイッチ	V1S/V1H V2
動作クロックスイッチ	8MHz /4MHz
ディスプレイスイッチ	24.8kHzタイプ /15.7kHzタイプ
CRTクロックスイッチ	内部クロック /外部クロック
	外部へ出力/ 外部から入力

4 セットアップモードによる初期設定

1. セットアップモードとは

セットアップモードとは，システム起動時の状態設定をメニュー形式で簡単に行うモードです．セットアップモードは，PC を押したまま電源スイッチをON，またはリセットボタンを押すことにより起動します．開梱時には下記のとおり初期値が設定されています．設定を変更する場合にのみ，セットアップモードを起動します．

本機では不揮発性RAMを利用したメモリスイッチを採用しており，システム起動時の状態設定を容易にしました(本章以降，セットアップモードで行う内容の設定をメモリスイッチと呼びます)．

セットアップモードの選択内容

• 初期設定

：開梱時の設定

項　目	選択内容	備　考
立ち上げモード	ターミナル/BASIC	N_{88}-BASICモードで有効
1行あたりの文字数	80字/40字	
1画面あたりの行数	25行/20行	
システムの立ち上げ	ディスク/ROM	開梱時の設定 PC-8801FH モデル10→ROM PC-8801FH モデル20 PC-8801FH モデル30 PC-8801MH （モデル20・モデル30・MH）→ディスク
内蔵FDD I/F	禁止/使用	開梱時の設定 PC-8801FH モデル10 PC-8801FH モデル20 PC-8801FH モデル30 PC-8801MH （以上）使用
メモリーウェイト	ON/OFF	
拡張スロットタイプ	タイプ1/タイプ2	PC-8801MHにはありません

• RS-232C設定

：開梱時の設定

項　目	選択内容
通信速度［ボー］	75　150　300　600　1200　2400　4800　9600　19200
通信方式	半二重/全二重
Xパラメータ	有効/無効
ストップビット長	2bit/1bit
データビット長	8bit/7bit
Sパラメータ	有効/無効
DELコード	有効/無効
パリティチェック	有り/無し
パリティ	偶数/奇数

2. セットアップモードの内容

PC を押しながら起動されるセットアップモードの内容は，PC-8801FHには16項目，PC-8801MHには15項目あります．それでは，各項目について簡単に説明します．

(1) 立ち上げモード

起動時のモードをBASICモードにするか，ターミナルモードにするかを設定します．

ターミナルモードについては，本書第17章を参照してください．

(2) 1行あたりの文字数

N88-BASICモード起動時の1行あたりの文字数を設定します．

(3) 1画面あたりの行数

N88-BASICモード起動時の1画面あたりの行数を設定します．

(4) システムの立ち上げ

システムの立ち上げを，フロッピーディスクから立ち上げるか，ROMから立ち上げるかを設定します．

(5) 内蔵FDD I/F

内蔵フロッピーディスクドライブインタフェースを使用するか，禁止するかを設定します．

(6) メモリーウエイト

メモリリードサイクルに1WAIT動作をさせるかどうかを設定します．

(7) 拡張スロットタイプ(この項目はPC-8801MHにはありません)

PC-8801mkⅡSRの拡張スロットバス3と同じ信号形態にするか，バス1，バス2と同じにするかを設定します．

(8) 通信速度(ボー)

通信速度を設定します．

通信速度については，本書第17章を参照してください．

(9) 通信方式

通信方式が全二重か半二重かを設定します．

通信方式については，本書第17章を参照してください．

(10) Xパラメータ

Xパラメータを有効にするか無効にするかを設定します．

Xパラメータについては，本書第17章を参照してください．

(11) ストップビット長

ストップビットの長さを1ビットにするか，2ビットにするかを設定します．

ストップビットについては，本書第17章を参照してください．

⑫ データビット長

データビットの長さを8ビットにするか，7ビットにするかを設定します．

データビットについては，本書第17章を参照してください．

⑬ Sパラメータ

Sパラメータを有効とするか無効とするかを設定します．

Sパラメータについては，本書第17章を参照してください．

⑭ DELコード

DELコードを受信したとき，処理するか無視するかを設定します．

DELコードについては，本書第17章を参照してください．

⑮ パリティチェック

パリティチェックの有無を設定します．

パリティについては，本書第17章を参照してください．

⑯ パリティ

偶数パリティにするか奇数パリティにするかを設定します．

3. セットアップモードの初期設定の変更方法

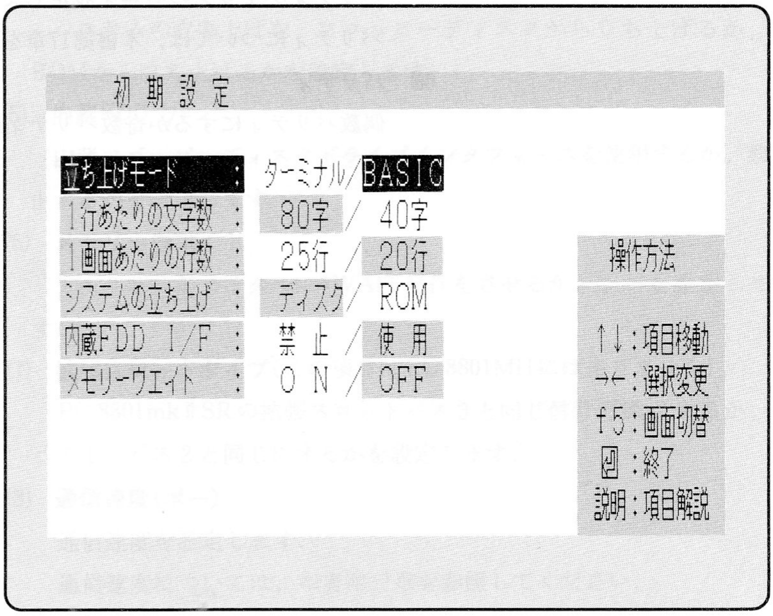
通常，PC を押しながら起動されるセットアップモードで初期設定を変更する必要はほとんどありませんが，場合によって（特に通信関係の設定）は変更する必要があります．

そこで，まずセットアップモードでの初期設定の変更方法を説明します．

(1) セットアップモードの起動方法

セットアップモードを起動するには，PC を押したままリセットボタンを押します（PC を押したまま電源スイッチをONにしても同様の機能が働きます）．このとき PC は，画面が表れるまで押し続けてください．

下図のような画面が表れます．



（PC-8801MHの場合）

(2) セットアップモードの操作方法

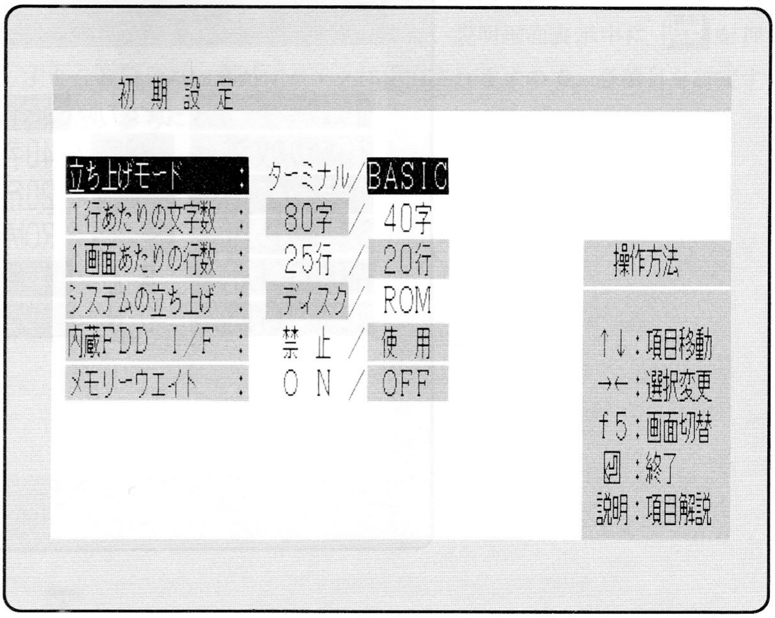
セットアップモードが起動したら，変更したい項目を設定しなおしましょう．

各項目のうち，現在カーソルがある行は赤地に黒文字で表され，その他の項目は水色地に黒文字で表されます．そして，それぞれの項目で現在選択されている選択肢が赤地，青地で示され，選択されていない選択肢は，ただの黒文字で示されます．また，カーソルは点滅しません．

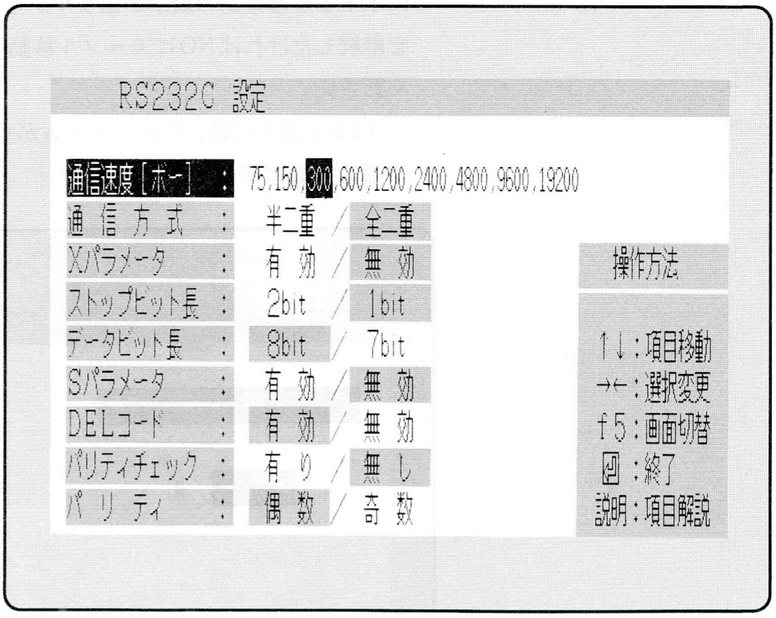
使用するキーは，↑，↓，→，←，⏎，f・5，説明の7つです．

f・5 は設定画面の切り替えに使います．設定画面は次ページの

図に示すように初期設定とRS-232C設定の２つがあります．



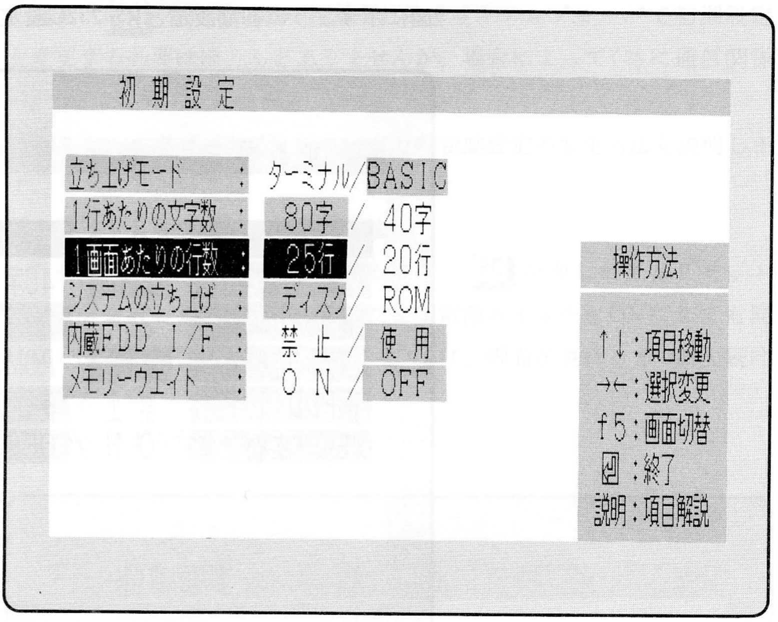
（PC-8801MHの場合）



カーソル移動キーは項目の移動，設定の変更に使います．

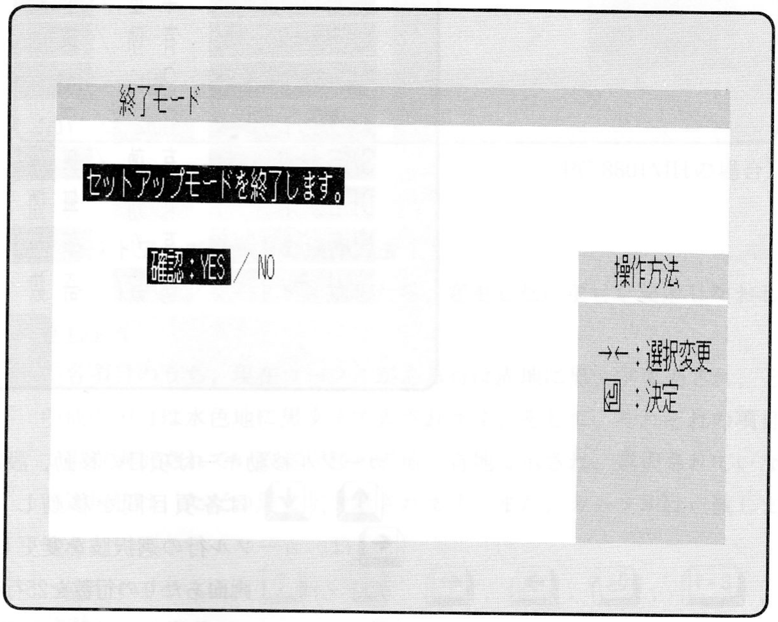
[↑]，[↓] は各項目間を移動し，カーソル行を変更し，[→]，[←]は，カーソル行の選択肢を変更します．

たとえば，１画面あたりの行数を25行に設定したい場合，[↑][↓]を使ってカーソル移動キーを"１画面あたりの行数"の項目に合わせ，[→][←]を使ってカーソル移動キーを25行のところに合わせます．



[↵] は，セットアップモードを終了します．[↵] を押すと，下図に示すような確認の表示が出ますから，よければYES，セットアップを継続したければNOにカーソル移動キーを合わせて [↵] を押してください．

YESを選んだ場合にはシステムが立ち上がり，NOを選んだ場合には設定画面に戻ります．



説明 は，各項目の簡単な説明を表示します．

セットアップモードのとき 説明 を押すと，現在カーソルがある項目の内容が画面に表示されます．また，説明画面表示中に 画面コピー を押すと，各項目の内容がプリンタに出力されますので，各項目を設定する場合，参考にしてください．

第3章

キーボード

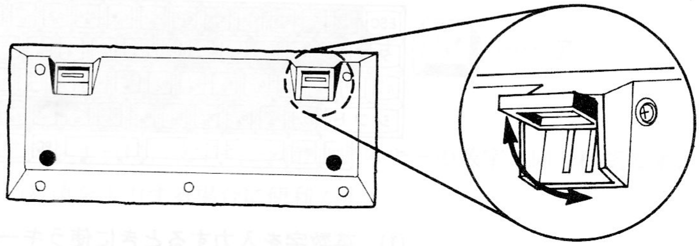
私たちが人と会話するとき，頭の中で理解/判断をし，相手の話に対して答えますね．
パソコンでも同じことがいえます．ただ，パソコンとの会話の場合，キーボードが私たちとパソコンを結ぶ“インタフェース”ですから，まずキーボードに触れなければ会話は成り立ちません．
また，キーボードに慣れることによって，パソコンへの親しみもわいてくることでしょう．
この章では，キーボードの外観，主なキーの働きを説明し，最後にキー入力のための基本的なことがらを説明します．

1 キーボードの外観/キーの配置

1. キーボードの外観

本機のキーボードは，本体とは分離し，カールケーブルによって結ばれ，自由な位置でキー入力できるようになっています．

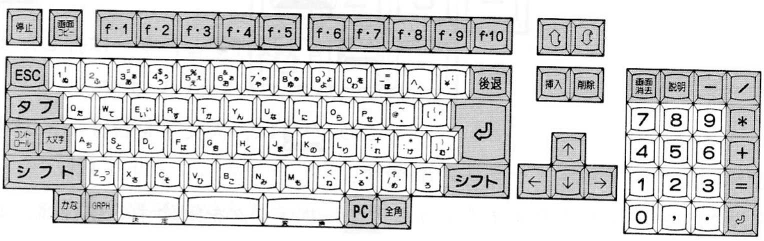
また，背面後部には折りたたみスタンドが組み込まれており，スタンドを立てると，キーが打ちやすい角度にボードを傾けることができます．



キーの角度やキートップの形もタッチ感覚を考慮して，各段ともそれぞれ角度を変化させてあり，キータッチ感覚の向上が図られています．

2. キーの配置

本機のキーボードは，下図に示すキー配置になっています．



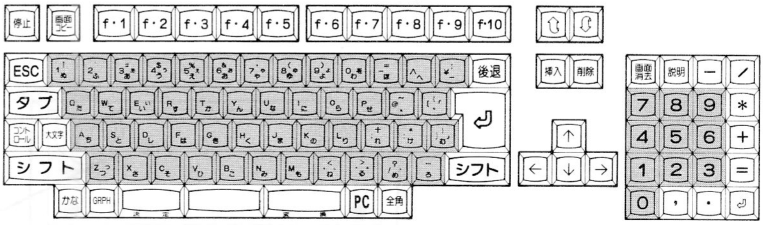
キーボードは2種類に色分けされていて，濃いブラウン(上のイラストでは　　　)に塗られたキーは特殊キーと呼び，他のクリーム色(上のイラストでは白)に塗られたキーは文字キーと呼びます．キーボードの右側には電卓のキーと似た数値入力用のキー(テンキー)があり，多くの数値を入力する場合に便利です．

2 キーボードの使い方

1. 文字キー

下図に示す▒▒が文字キーです．これらのキーを押すと，対応する文字がディスプレイに表示されます．

文字キーの入力方式には，英数字モード，カナモード，グラフィックキャラクタモード，日本語入力モードの4つのモードがあります．



(1) 英数字を入力するときに使うキー

アルファベットや数字，"＋"，"＊"，"？"などの特殊記号などの入力モードです．単独でキー入力した場合は，キートップの中段左側に書いてある文字が入力されます．アルファベットは，タイプライタと同様に小文字が入力されます．

[A ち] ……a　　[3 # ぁ あ] ……3

[大文字] を1度押すと，キーは押したままの状態に固定されます．これを，ロックするといいます．

この [大文字] は，大文字を続けて入力する場合に使用すると大変便利です．

[大文字] のロックを解除する場合は，もう1度 [大文字] を押します．

(2) カタカナを入力するときに使うキー

カタカナや"「" "、" "。"などの和文の特殊記号を入力するモードです．

カナモードは，[かな] をロックした状態でキー入力します．

この場合，キートップの下段に書いてある文字が入力されます（キートップの表記はひらがなですが，入力されるのはカタカナです）．

[かな]，[A ち] ……チ　　[かな]，[3 # ぁ あ] ……ア

(3)　シフトキー [シフト]

シフトキーは左右に２つありますが，どちらも同じ働きをします．

①　かなキーがロックされていないとき

かなキーがロックされていないとき [シフト] を押しながら文字キーを押すと，キートップの上段の文字やアルファベットの大文字が入力されます．

[シフト] ＋ [A ち] ……A　　[シフト] ＋ [3 # ぁ あ] ……#

また，[大文字] をロックした状態で [シフト] を押しながら入力すると，小文字が入力されます．

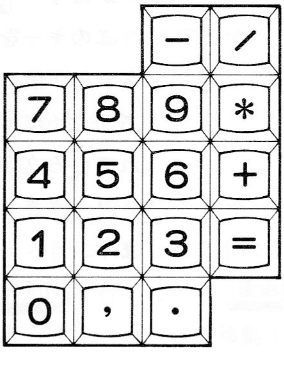
②　カナモードのとき

カナモードのときに [シフト] を押しながらキー入力すると，キートップの中段右側に書いてある文字が入力されます．

[かな]，[シフト] ＋ [3 # ぁ あ] ……ア

(4)　テンキー

テンキーは，文字キーの最上段のキーの数字と同じ機能です．多くの数値を入力する場合に便利です．

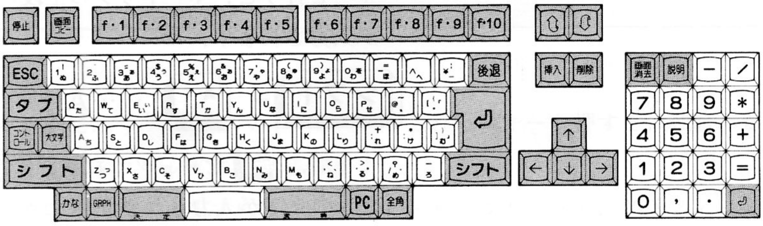


英数字モード，カナモード，日本語入力モードではテンキーに対応する数字や記号が入力され，グラフィックキャラクタモードではグラフィックキャラクタが入力されます．

2. 特殊キー

下図に示す　　　のキーが特殊キーです．

これらのキーは，使用するソフトウェアによって機能や意味が異なります．具体的なキーの機能については，それぞれのソフトウェアの解説を参照してください．ここでは特殊キーの代表的な機能を紹介します．



(1) リターンキー [↵] [↵]

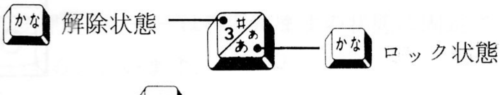
[↵] と書かれたキーが [後退] の下にあります．リターンキーと呼ばれています．キーボード右側にも，大きさは違いますが [↵] と書かれているキーがあります．どちらもコンピュータに対して入力があったことを知らせる働きをします．N88-BASICなどのコマンドはこのキーを押さなければ実行されません．

(2) モード変換

① かなキー [かな]

このキーを押すとキーはロックされ，入力はカナモードになり，キーの下段に書かれているカナ文字が表示されます(ただし，ディスプレイにはカタカナで表示されます)．また，キー右側に表示されている文字は，[シフト] を押しながら入力すると表示されます．

ロックされた [かな] をもう１度押すと，[かな] のロックは解除されます．



② グラフィックキー [GRPH]

このキーを押しながら文字キーを押すと，入力はグラフィックキャラクタモードになり，グラフィックキャラクタや"年"，"月"などの簡単な漢字を表示・入力することができます．どのキーがどのキャラクタにあたるかは，「ユーザーズガイド」の巻末資料をご参照ください．

グラフィックキャラクタモードは，N88-日本語BASICでは使用できません．

③ 大文字キー [大文字]

このキーを押してロックすると，それ以後，アルファベットを入力すると大文字が入力されます．このとき，[シフト] を押しながら入力すると小文字が入力されます．[大文字] のロックを解除する場合

は，もう1度このキーを押します．

(3)　シフトキー [シフト]

ファンクションキーの内容の表示を変えたり，他の特殊キーと組み合わせて，カーソルをホームポジションに移動させたり，ターミナルモードからBASICモードに切り替えたりします．

(4)　コントロールキー [コントロール]

他のキーと組み合わせて，特殊な機能のコードを入力します．他の文字キーとの組み合わせとその機能はソフトウェアによって異なりますので，それぞれの解説を参照してください．

(5)　編集キー

①　カーソル移動キー [↑] [↓] [←] [→]

カーソルを上下左右の方向に移動させるキーです．

②　削除キー [削除]

カーソルの直前の文字を1文字消します．カーソルの後に文章が続いている場合には，それらの文字がすべて前へ詰められます．カーソルより後方へ向かって文字を消去するのに使用します．次の[後退]と同じ働きをします．

③　後退キー [後退]

カーソルの直前の文字を1文字消します．カーソルの後に文章が続いている場合には，それらの文字がすべて前へ詰められます．カーソルより後方へ向かって文字を消去するのに使用します．前の[削除]と全く同じ働きをします．

④　挿入キー [挿入]

N_{88}-BASICなどの文字入力中にこのキーを押すと，インサートモードに変わり，その後に入力した文字はすべてカーソルの直前に挿入されます．[シフト]を押しながら[後退]（[削除]）を押してもインサートモードになります．カーソル移動キーを押したり，もう1度[挿入]を押すとインサートモードからぬけ出します．

⑤　タブキー [タブ]

カーソルを移動させるキーです．ただし，このキーを使ったときのカーソル移動の仕方は，それぞれのソフトウェアで異なります．

ソフトウェアによっては，メニュー項目の選択に使います．

⑥　ロールアップ／ロールダウンキー [⇧] [⇩]

[f・9]（[シフト]＋[f・4]）を押したときや，EDITコマンドを実行したときにおいて，画面に表示されている行の前後を表示するときに使用します．

⑦　画面消去キー [画面消去]

画面に表示されている内容を消し（クリアされるといいます），

カーソルは左上スミ(ホームポジションと呼びます)に移動します.

消す内容は，N88-BASICにおいては文字表示のみですが，N88-日本語BASICにおいては文字表示とグラフィックの両方を消します.

［シフト］を押しながら［画面消去］を押した場合は，現在画面に表示されている内容はそのままで，カーソルだけが画面の左上スミ(ホームポジション)に移動します(消えるのは画面表示だけで，メモリ上のプログラムは消えません).

(4) 日本語入力に使うキー

① 全角キー ［全角］

N88-日本語BASICのときに意味を持つキーです．このキーを押すと全角文字入力モードになり，日本語の漢字やひらがななどの全角文字を入力することができます(N88-BASICのときには意味を持ちません).

もう1度押すと，全角文字入力モードが解除されます.

② 変換キー ［変換］

入力した文字を漢字に変換します．ただし，このキーはN88-日本語BASICのときに意味を持ち，N88-BASICのときは［　］を意味します.

③ 決定キー ［決定］

変換キーで変換された文字を決定したり，ひらがななどを変換せずに決定します．このキーは，N88-日本語BASICのときに意味を持ち，N88-BASICのときは［　］を意味します.

(5) ファンクションキー ［f･1］ ～ ［f･10］

このキーは，あなたのキー操作を簡略化し，手助けするために設けられたものです.

各ソフトウェアでキーの内容は異なります．また，ユーザが自分で定義することもできます.

CONSOLE文の第3パラメータを"1"に設定した場合(開梱時の初期設定は1です)，［f･6］～［f･10］を表示するには［シフト］と同時に［f･1］～［f･5］を押します．N88-日本語BASICでCONSOLE文の第3パラメータを"2"に設定した場合，［f･6］～［f･10］をそのまま押せば表示できます.

(6) 停止キー [停止]

プログラムの実行などを中断するときに使います．また [シフト]＋[停止] でターミナルモードからBASICモードに切り替えることができます．

(7) エスケープキー [ESC]

このキーの使い方はソフトウェアによって異なりますので，それぞれの解説を参照してください．

(8) 画面コピー [画面コピー]

ディスプレイに表示されている画面をプリンタに印字します．すなわち画面の表示内容を紙の上にコピーしたいときにこのキーを使います．

セットアップモードのとき [画面コピー] を押すと，各項目の内容がプリンタに出力されます．

もしプリンタが接続されていない場合は，このキーは意味のないものになります．押してしまったときは，[停止] を押して中断してください．

(9) 説明キー [説明]

プログラム実行中にエラーが発生したときに [説明] を押すと，エラー発生行が表示され，簡単にプログラムを修正できます．

また，セットアップモードのとき [説明] を押すと，現在カーソルのある項目の内容が画面に表示されます．

(10) セットアップモードの起動 [PC] ＋リセットボタン

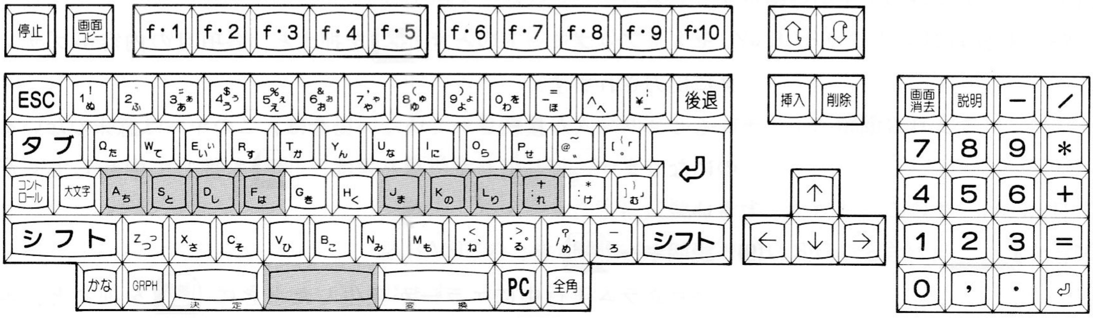
[PC] を押したままリセットボタンを押すと，セットアップモードが起動します（[PC] を押したまま電源スイッチを投入しても同じ機能が働きます）．

[PC] は，セットアップモードの起動時以外のときに押されても無視します．

3 ホームポジション

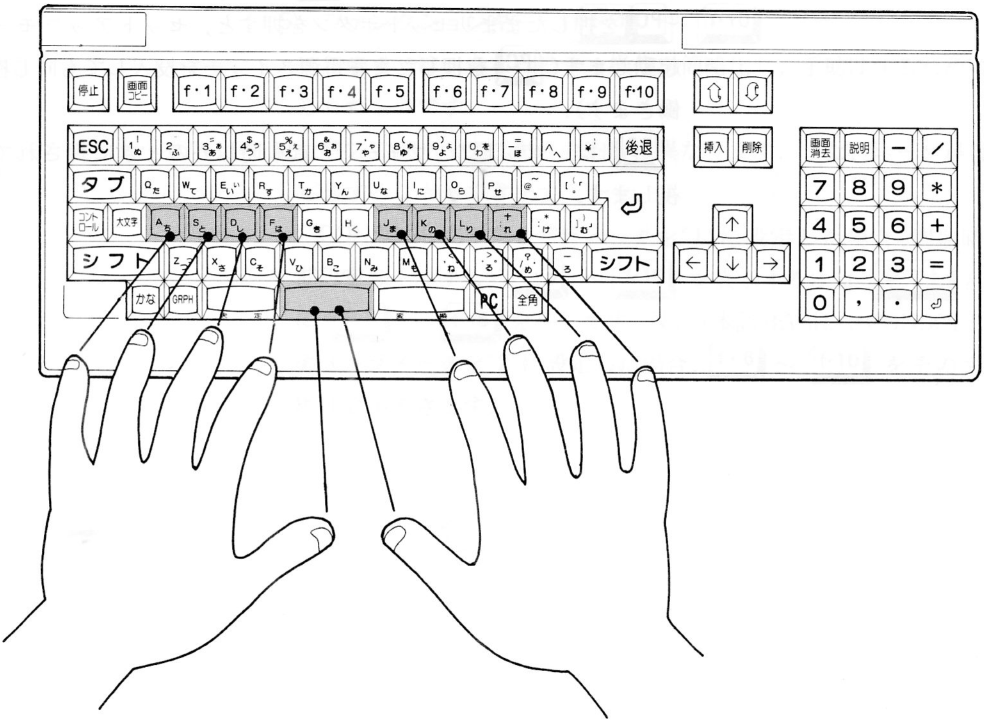
英文タイプの経験がある方ならばご存知と思いますが，キーによって使う指が決められています．この基点となるのがホームポジションです．

下図に示す　　　　　がホームポジションです．



このホームポジションを基点にして各キーの位置を推測してキータッチの練習をしてください．

下図は各キーに対応する指を示します．



第4章

スクリーンエディタ

この章では英数字だけを使って，スクリーンエディタの基本的な使い方を説明していきます．
N88-BASICおよびN88-日本語BASICでは，プログラムを作ったり，コマンドやステートメントをダイレクトに実行させたりするときにスクリーンエディタという画面編集機能を使います．
スクリーンエディタは，ディスプレイ画面上のどの位置にでもカーソルを移動して，入力，修正をすることができます．
これによって，修正する行を改めて最初から入力する必要がなくなり，プログラムの入力や修正も短時間で行うことができます．
N88-日本語BASICでの日本語文字を混じえたスクリーンエディタの使い方は，第5章をご覧ください．

1 スクリーンエディタに入る前に

文字などを入力する前に，必ず覚えておかなければならない項目(用語)がいくつかあります．ここでは，その項目(用語)について説明します．

1．"キー入力が可能な状態"とは？

N_{88}-BASICを起動させると，初めに

```
How many files(0-15)? █
```

というメッセージが画面に表示されます．

ここで  を押してください．そうすると，

```
How many files(0-15)?
NEC N-88 BASIC Version 2.3
Copyright (C) 1981 by Microsoft
***** Bytes free
Ok
```

となります．

この上図のメッセージが画面に表示されれば，いつでもキー入力が可能です．

注意 この"How many files(0-15)?"の意味については本書の第2章を参照してください．

2．"カーソル"とは？

先ほどの画面表示で"Ok"の下で点滅(ブリンクといいます)していた長方形のマークのことを"カーソル"といいます．

このカーソルの位置に，キーボードによって文字などが入力できます．したがって，カーソルを，カーソル移動キーでコントロールすることにより，画面上の任意の位置に文字などを入力することができます．

またカーソルは，文字などが入力されると自動的に隣の桁へ移動します．

3. "全角/半角"とは?

入力できる文字には，全角文字と半角文字があります．全角文字で入力する場合は，漢字を扱う日本語入力のときです(詳しくは第5章を参照してください)．それ以外のときは半角文字で入力します．

半角文字と全角文字の入力モードは，[全角]を押すことによって切り替えられます．

4. " [↵] "とは?

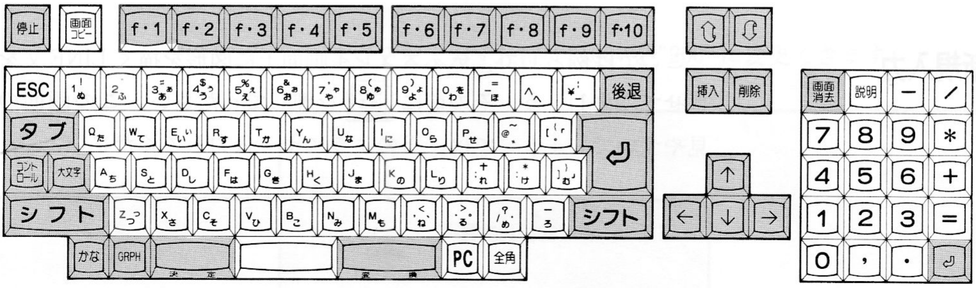
文字を画面にただ入力しただけでは，入力された文字が画面に表示されるだけで，コンピュータは何の働きも起こしません．

コンピュータに，「命令(または，データ)を受け入れなさい」という命令を送る必要があるのです．この命令を伝える役割を受け持っているのが[↵]というキーです．

また，[↵]を押してコンピュータを働かせることを"実行させる"といいます．

2 編集に使うキー

編集のために使うキーにはいろいろな種類がありますので，ここで整理してまとめて記載します．使い方を忘れたときには本節を参考にしてください．



編集キー	機能
↑ ↓ ← →	カーソルを上下左右に動かします．
↵	コンピュータにデータ/命令の入力があったことを知らせる働きをします．
後退	単独で押すとカーソルの前にある文字を１文字削除します． シフト を押しながらインサートモードに入るとき，抜けるときに使用します．
挿入	インサートモードに入るとき，抜けるときに使用します．
削除	カーソルの前にある文字を１文字削除します．
画面消去	単独で押すとテキスト画面(文字)を消去します． シフト を押しながら押すとカーソルを画面の左上（ホームポジション）に移動します．
f·1 ～ f·10	キーの内容をユーザが定義することもできます．
シフト	アルファベットの大文字，キートップの上段に書いてある特殊文字の入力に使用します．
大文字	シフト を押さなくてもアルファベットの大文字が入力できるようになります．
かな	仮名を入力するときに使います．ただし，画面上で，半角入力モードの場合，カタカナが表示されます．
GRPH	グラフィックシンボルの入力に使用します．
停止	プログラムの中断に使います．
タブ	カーソルを水平方向に８桁ずつ移動します（そのとき，スペースが入力されています)．
コントロール	後述のコントロールコードの入力に使用します．
⇧ ⇩	後述のエディットモードにおいて，画面に表示されている行の前後を表示するときに使用します．
変換	後述の日本語入力モードにおいて，かな漢字変換に使用します．
決定	後述の日本語入力モードにおいて，かな漢字変換された文字を決定します．

3 ダイレクトモードにおけるスクリーンエディタ

コマンドやステートメントをプログラムにしないで単独で実行させる使い方を，ダイレクトモードといいます．

1．新規入力

ここでは例として，ディスプレイ画面上に図形を描くLINE文を実行させてみましょう．その前に[画面消去]を押してください．この画面の方が見やすいでしょう．では，下のようにキー入力してください．

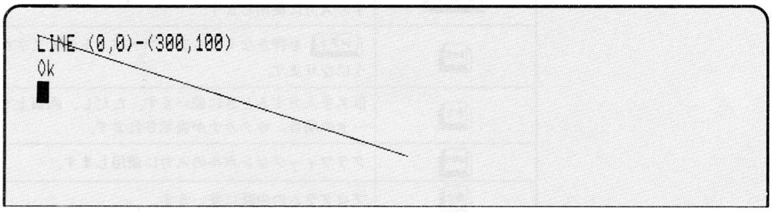
```
LINE (0,0)-(300,100)█
```

[大文字]がロック(押された状態になっていること)されていれば大文字，そうでなければ小文字が入力されますが，どちらでもかまいません．

[かな]がロックされているとカタカナが入力されてしまいますから，ロックを解除してから入力してください．

ここで[↵]を押してLINE文を実行させてみますが，もし，その前に入力の間違いに気がついたら，カーソルを間違っている位置に移動して入力しなおしてください．

では，[↵]を押してください．



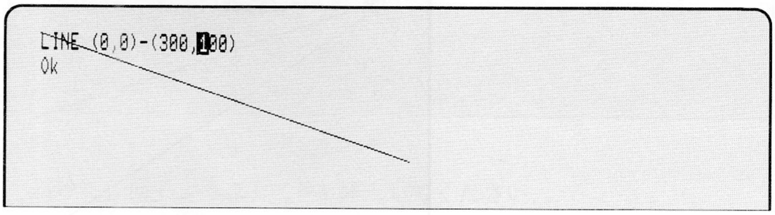
LINE文が実行されて画面に直線が描かれました．

エラーが出てしまった場合は，もう1度正確に入力しなおしてみてください．画面に何も表示されない場合は，リセットボタンを押して，もう1度BASICの起動からやりなおしてください．

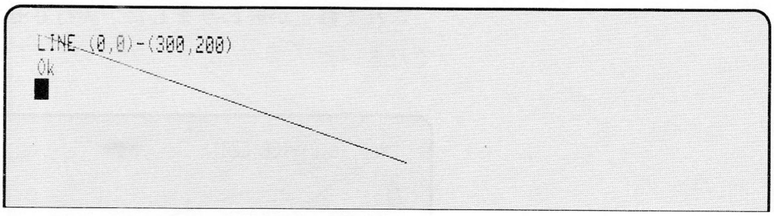
2. 訂正

スクリーンエディタを使って，LINE(0, 0)-(300, 100)をLINE(0, 0)-(300, 200)になおして実行してみます.

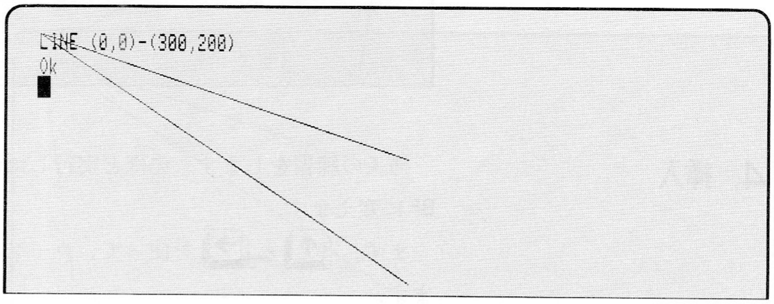
[↑]と[→]を使って，カーソルを下の図のように"1"の位置まで移動させます.



ここで2を入力すれば，"100"を"200"に変更できます.

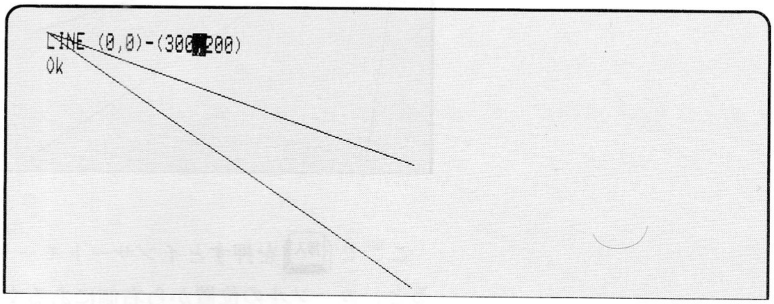


実行させてみましょう. [↵]を押してください.



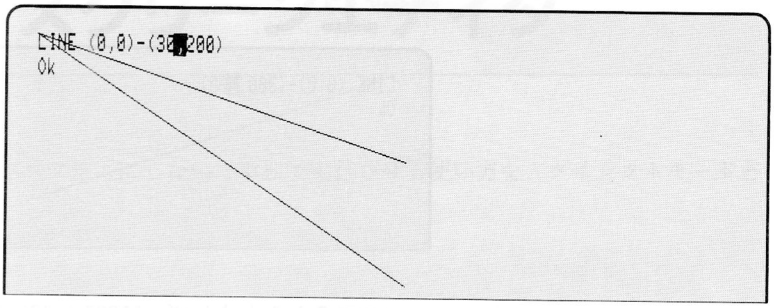
3. 削除

次に，LINE(0, 0)-(300, 200)をLINE(0, 0)-(30, 200)になおして実行してみます.

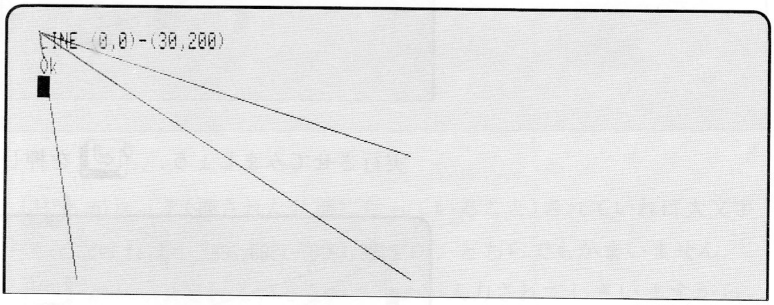


[↑]と[→]を使って，カーソルを下の図のように","(カンマ)の位置まで移動させます．

0を1つ削除するには，ここで[削除]を押します．



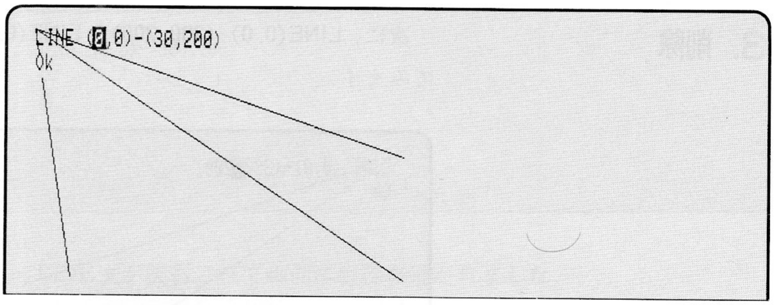
これで修正が終わりました．実行させてみましょう．[↵]を押してください．



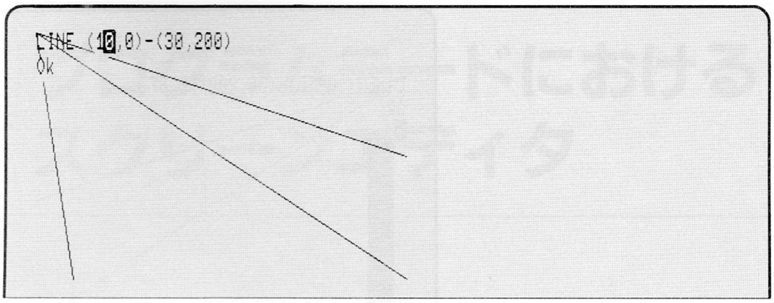
4. 挿入

挿入の練習をします．先ほど実行した命令をLINE(10,50)-(30,200),2,BFに変えます．

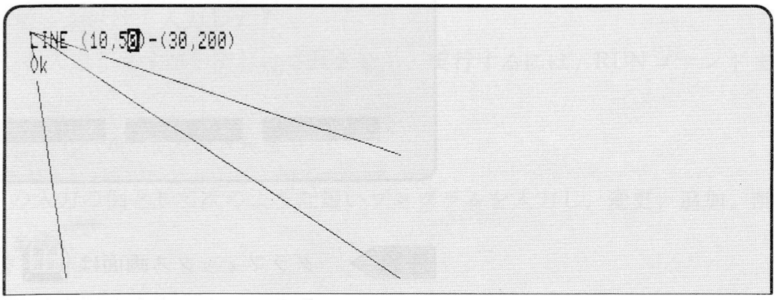
まず，[↑]と[→]を使って，カーソルを下の図の位置まで移動させます．



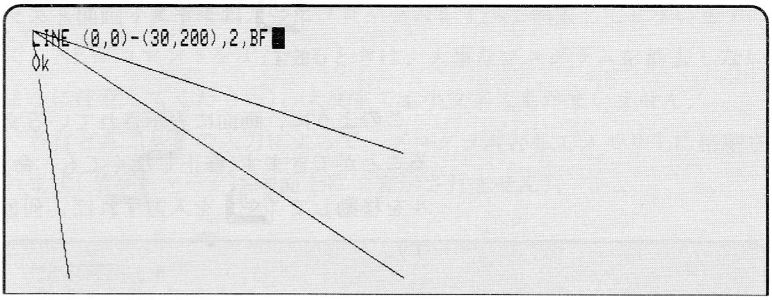
ここで[挿入]を押すとインサートモードになります．1[↵]を入力すると，カーソルの位置から右側にある文字が1つ右にずれていきます．



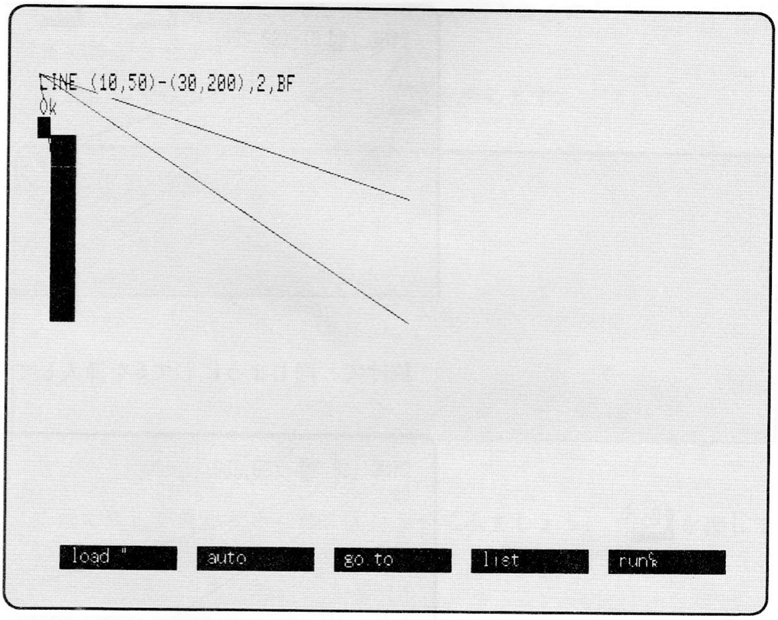
続けて，同じようにして5を挿入してください．



そして，命令の最後に",2,BF"を追加します．



修正が終わりましたので，を押せば実行されます．



注意 グラフィックス画面は［画面消去］を押しても消去されません．グラフィックス画面を消去したいときは，CLS␣2［↵］と入力してください．

また，CLS␣1［↵］はテキスト画面（文字）のみ，CLS␣3［↵］はテキスト画面とグラフィックス画面の両方を消去します．

このように，画面に表示されている文字は修正して何度でも実行させることができます(修正しなくても，命令の表示されている行にカーソルを移動して［↵］を入力すれば，何回でも実行させることができます)．

4 プログラムモードにおける スクリーンエディタ

BASICのコマンドやステートメントの集まりを続けて実行させるときは，プログラムを作り，RUNコマンドによって実行します．プログラムを入力するモードをプログラムモードといい，前述のダイレクトモードとは次の点が異なります．

- 文の先頭に1～65529までの数字を付けて入力します．
- ⏎ で1行(255文字以内)を入力しますが，実行はされません．実行するには，RUNコマンドまたはGOTO文を使います．
- ⏎ によって1行を入力しても"Ok"は表示されません．

ここでは，プログラムモードの入力の例として次のような短いプログラムを入力し，変更，追加，削除などの方法を具体的に説明します．

1. プログラムの新規作成

まずNEW ⏎ とキー入力し，メモリ上にはプログラムが何もない状態にしておいてから，次のプログラムを入力します(ただし，NEWコマンドは，今までに入力したプログラムをすべて消去してしまいますので，NEWコマンドを実行するときは，大事なプログラムを消去しないように注意してください)．大文字でも小文字でもかまいません．

各行とも ⏎ の入力によって，プログラムとしてメモリ上に格納されます(⏎ マークは画面上には表示されません)．

```
10 SCREEN 0,0
20 FOR I=224 TO 247
30 PRINT CHR$(I);
40 NEXT I
50 END
█
```

プログラムを新しく入力したり変更などをした場合，必ずLISTコマンドで正しく入力されたかどうかを確かめます．LIST ⏎ と入力すると，次ページのように表示されます．

```
list
10 SCREEN 0,0
20 FOR I=224 TO 247
30 PRINT CHR$(I);
40 NEXT I
50 END
Ok
█
```

もし，LISTコマンドによって表示されたプログラムに誤りがあったら，ダイレクトモードの項で述べたようにして訂正します．このときも ↵ を入力しなければ，何も変更したことになりません．プログラムを変更するときには，必ず ↵ を入力しなければなりません．

RUN ↵ と入力すると，下のプログラムが実行されます．

```
list
10 SCREEN 0,0
20 FOR I=224 TO 247
30 PRINT CHR$(I);
40 NEXT I
50 END
Ok
run
=┣┫┣┫▲▼♠♥♦♣●○／＼╳円年月日時分秒
Ok
```

もし，エラーメッセージが表示されたら，LISTコマンドでプログラムを表示し，カーソルを動かしてエラーのあった部分を修正し， ↵ を押してプログラムを変更すればよいのです．

スクリーンエディタの機能を用いてプログラムの訂正を行った場合は，各行番号の訂正が終わるたびに，必ずリターンキー（ ↵ ）を入力してください．

2. 行の挿入と追加

BASICのプログラムは，行番号の小さいものから順に実行されます．

(1) 行の挿入

先ほどのプログラムで，行番号20と30の間に新しく行を挿入する場合は，21〜29の行番号を付けた行を新しく入力すればよいのです．

行番号を25として下のように入力してみます．

```
25 BEEP█
```

プログラムの変更を行いましたから，LISTコマンドを実行して，正しく入力されているか確かめてみます．

```
list
10 SCREEN 0,0
20 FOR I=224 TO 247
25 BEEP
30 PRINT CHR$(I)
40 NEXT I
50 END
Ok
█
```

RUN ⏎ と入力して実行させてみましょう．もしエラーが出たら，カーソルを誤りのある部分に移動してなおし，⏎ を押します．

(2)　行の追加

同じく先ほどのプログラムで行番号40の後に追加する場合は，41以上の行番号を付けます．

3. 行番号の整理

行をいくつか挿入して行番号が混み合ってきたときなどには，RENUMコマンドを使って行番号を整理します．下のようにRENUMコマンドを実行し，LISTコマンドでプログラムを表示してみましょう．

```
RENUM 1000
Ok
list
1000 SCREEN 0,0
1010 FOR I=224 TO 247
1020 BEEP
1030 PRINT CHR$(I)
1040 NEXT I
1050 END
Ok
█
```

RENUMコマンドは，各行番号の間の増分や，プログラムのどこから行番号を付け替えるかの指定もできます．詳しくは，「リファレンスマニュアル」をご覧ください．

4. 行の複写

1つのプログラムの中に，内容が同一の行を何箇所か作る場合は，行番号の部分だけを書き換えるだけで行のコピーができます．

たとえば，先ほどのプログラムに次ページのような2行を追加したい場合は，行番号1010と1040のコピーを作ればよいわけです．

```
10 FOR I=224 TO 247
20 NEXT I
```

カーソルを下図の位置に移動し，[削除]を2回押して [↵]を押してください．

```
list
1000 SCREEN 0,0
1010 FOR I=224 TO 247
1020 BEEP
1030 PRINT CHR$(I)
1040 NEXT I
1050 END
Ok
```

同じように**1040**を**20**と書き換えて [↵]を押します．

```
list
1000 SCREEN 0,0
10 FOR I=224 TO 247
1020 BEEP
1030 PRINT CHR$(I)
1040 NEXT I
1050 END
Ok
```

画面上からは行番号**1010**と**1040**はなくなってしまいましたが，メモリ上には残っています．LISTコマンドを実行してプログラムを表示させてみましょう．

```
list
10 FOR I=224 TO 247
20 NEXT I
1000 SCREEN 0,0
1010 FOR I=224 TO 247
1020 BEEP
1030 PRINT CHR$(I)
1040 NEXT I
1050 END
Ok
█
```

5. 行の削除

(1)　1行の削除

1行ずつの削除は，行番号だけを入力して [↵] を押します．たとえば，先ほどのプログラムで下のように入力すれば，行番号1020が削除されます．

```
1020
```

LISTコマンドを実行してプログラムを表示させてみましょう．

```
list
10 FOR I=224 TO 247
20 NEXT I
1000 SCREEN 0,0
1010 FOR I=224 TO 247
1030 PRINT CHR$(I)
1040 NEXT I
1050 END
Ok
█
```

(2)　まとめて削除

プログラムのある範囲をまとめて削除する場合は，DELETEコマンドが便利です．たとえば，先ほどのプログラムで下のように行います．

```
DELETE 10-20
```

LISTコマンドを実行してプログラムを表示させてみましょう．

```
list
1000 SCREEN 0,0
1010 FOR I=224 TO 247
1030 PRINT CHR$(I)
1040 NEXT I
1050 END
Ok
█
```

プログラム全体を削除するときは，NEWコマンドを使用します．

5 便利なエディット機能

前の項までで，スクリーンエディタの基本的な使い方について説明しました．ここでは，プログラム作りに慣れたあと，より能率的にスクリーンエディタを使いたいときに便利な機能を紹介します．

1．コントロールコード

(1)　1行消去　[コントロール] ＋ [U]

カーソルが点滅している行を消去する場合は，[コントロール] ＋ [U] を入力します．その行の先頭から，行の最後までを画面から消去します．この場合は行番号も消えてしまいますから，[↵] を入力してもプログラムは訂正されません．

(2)　カーソルから，その行の終わりまでの消去 [コントロール] ＋ [E]

カーソルが点滅している文字からその行の終わりまで消去する場合は，[コントロール] ＋ [E] を入力します．この後で [↵] を入力すれば，メモリの内容も訂正されます．ですから行番号のすぐ次の位置にカーソルを移動し，[コントロール] ＋ [E] を入力し，その後 [↵] を入力すると，その行番号の文はプログラムから削除されます．

(3)　カーソルの移動　[コントロール] ＋ [F]，[コントロール] ＋ [B]，[コントロール] ＋ [X]

キーボードにはオートリピート機能が備わっているため，カーソル移動キーを押し続けると，カーソルは次々と移動します．しかし，さらに便利な機能として，項目ごとの移動ができます．

[コントロール] ＋ [F] を入力すると，カーソルは次のワード（単語）の先頭へ移動します．また [コントロール] ＋ [B] を入力すると，1つ前のワード（単語）の先頭へ移動します．また，[コントロール] ＋ [X] を入力すると，その行の最後の文字の次にカーソルが移動します．

(4)　行の分割とラインフィードコードの挿入　[コントロール] ＋ [J]

1行の内容を2行にわたるように書きなおしたい場合には，区切りたい箇所でカーソルを止め，[挿入] でインサートモードに切り替え [コントロール] ＋ [J] を押すと，カーソルの後ろからその行の最後までが次へ移動します．移動した行の先頭に行番号を付加し [↵] を押せば，これまでの1行の内容が2行にわたって書かれます．

また，インサートモードになっていないときに [コントロール] ＋ [J] を入力すると，その行の最後にラインフィードコード（LF）が挿入されます．

(5)　1項目の削除　[コントロール]＋[DEL]

カーソルの点滅している文字から，その項目の終わりまでを消去します．

2. ロールアップとロールダウン

プログラムが数十行にもわたる場合には，画面上にはその一部しか表示することができませんが，ロールアップ／ロールダウン機能を使うと編集はずっと楽にできます．

この機能は，巻き物を見るように，画面に表示されている行の前後を順に表示させ，修正する部分を探すために使います．これにより，訂正したい部分を短時間で探すことができます．

(1)　**エディットモード**

ロールアップ／ロールダウン機能は，通常のプログラムモードから，さらにエディットモードに移って使用します．以下の2つの場合にエディットモードに移ります．

①　エラーによってプログラムが止まり，エラーの含まれる行が画面に表示されているとき(すべてのエラーが，このように表示されるわけではありません)．

```
run
Illegal function call in 100
Ok
█
```

②　EDITコマンドを実行したとき．

たとえば，プログラムを実行して上のようになったとき，EDIT□.　[↵]と入力すると下のように表示されます([f･9]を押しても同じ働きをします)．

```
EDIT .
```

```
100 CIRCLE (X,Y),R
```

(2) **ロールアップ／ロールダウン**

①または②の状態で前の行が見たい場合は [⇩] を入力します．このキーを押し続けると前の行が次々と表示されていき，画面に入りきらない行は画面から消えてしまいます．しかし，エディットモードに入っているときは，[⇧] を入力することにより，画面から消えてしまった行を再度表示させることができます．スクリーンエディタの機能を利用して行を訂正したあと，[↵] を入力してください．

(3) **エディットモードの終わらせ方**

エディットモードから抜け出す場合は，[画面消去] を入力します．

第5章

日本語入力

「N88-日本語BASICシステムディスク」のフロッピーディスクで起動したBASICの状態を，日本語モードといいます．この日本語モードは，半角文字が入力できるモード(半角文字入力モードといいます)と全角文字が入力できるモード(全角文字入力モードといいます)の2つのモードを持っています．
プログラムの入力や実行などコンピュータの通常の働きは，半角文字入力モードのときに実行します．
そして，全角文字入力モードのときには，漢字が入力できます(英数字なども全角文字で入力されます)．ただし，コンピュータの通常の働きは，実行できません．
この章では，日本語モードのときの入力の方法を説明します．

1 日本語入力機能の特長

N88-日本語BASICの日本語入力機能の特長をまとめます．

1. PC-8801MHの場合

- JIS第1水準および第2水準の文字が使用できます．
- カナ入力とローマ字入力のどちらも使えます．
- 熟語または文節単位でカナを漢字混じりの語に変換することができます．文法処理を行い，接頭語，接尾語を含んだ文節を一度に変換できます．また，アラビア数字を漢数字に変換することもできます．

  9900→九千九百

- 学習機能のある，約4万語の辞書をシステムディスク上に持っていますから，使用頻度の高いものから順に変換候補の順序が更新されていきます．
  使用していくに従って，ユーザに使いやすい辞書になっていくわけです．

2. PC-8801FHの場合

- JIS第1水準および第2水準の文字が使用できます．
- カナ入力とローマ字入力のどちらも使えます．
- 熟語単位でカナを漢字混じりの語に変換することができます．
- 学習機能のある，約3万5千語の辞書をシステムディスク上に持っていますから，使用頻度の高いものから順に変換候補の順序が更新されていきます．
  使用していくに従って，ユーザに使いやすい辞書になっていくわけです．

注意 N88-日本語BASICシステムディスクにライトプロテクトシールが貼ってあるときは，辞書の学習機能は働きません．辞書を更新しながら使いたいときは，ライトプロテクトシールをはがして使ってください．

2 日本語モードの概要

日本語モードでは，

- 全角かな漢字(ひらがなと漢字)
- 全角カタカナ
- 全角英数文字

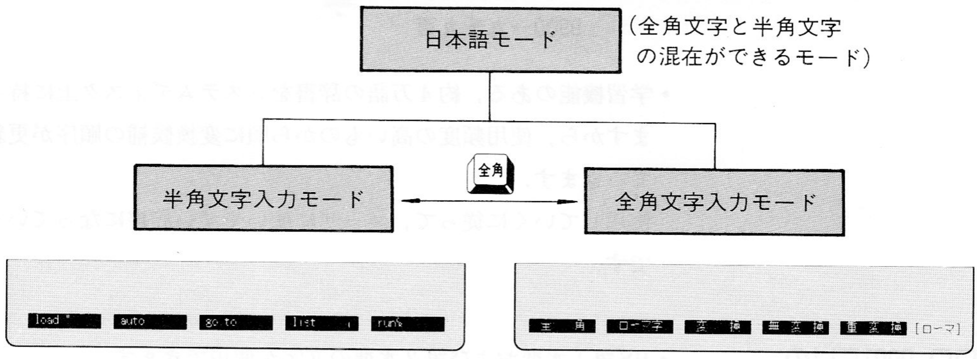
} 全角文字入力モード(プログラムの入力や実行が行えない)

- 半角カタカナ
- 半角英数文字

} 半角文字入力モード(プログラムの入力や実行が行える)

の５種類の文字が入力できます．

まとめると，下図のようになります．

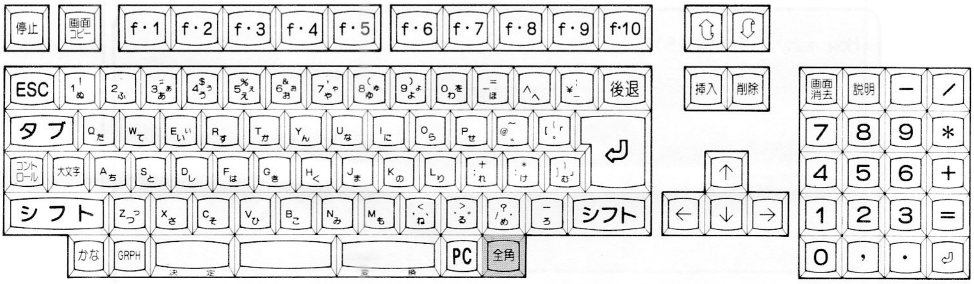


すべてのファンクションキーはユーザが定義できます．

すべてのファンクションキーは，全角文字の入力に使われているため，固定されています．

3 全角文字入力モードと半角文字入力モードの切り替え

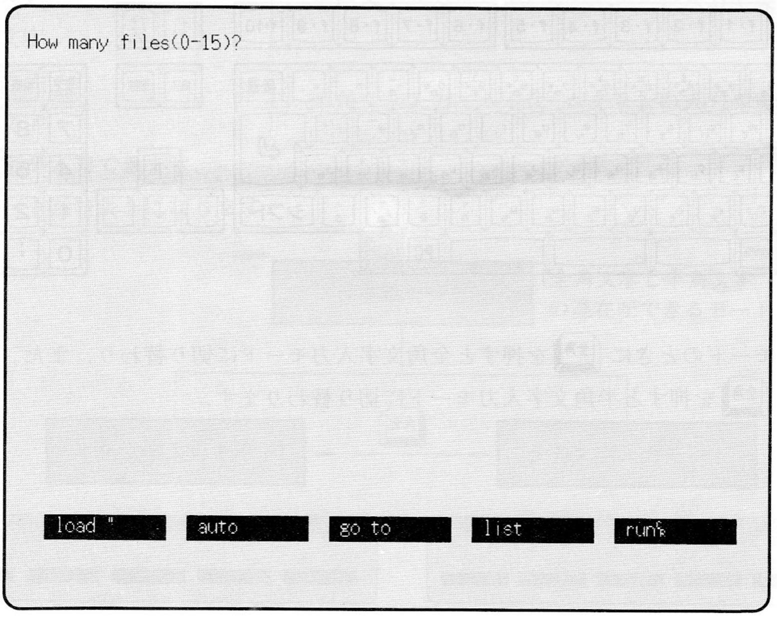
全角文字入力モードと半角文字入力モードの切り替えは，キーボードの一番下の左側から７番目の［全角］を押すことによって行います．



半角文字入力モードのときに［全角］を押すと全角文字入力モードに切り替わり，また，全角文字入力モードのときに［全角］を押すと半角文字入力モードに切り替わります．

4 日本語モード時の画面

日本語モードのときの初期モード(BASICが起動した直後のモード)は，半角文字入力モードです．下図のような画面が現れますが，この画面はN_{88}-BASICと同じ画面です．



これは80文字／行のときの表示例です．

ここで を押すと，

```
N88-日本語BASIC**** Version 2.0
Copyright (C) 1981 Microsoft Corporation
Copyright (C) 1985,1986 日本電気株式会社
使用可能メモリ容量：*****バイト
Ok



load "   auto   go to   list   run&
```

という日本語のメッセージが表示されます．この画面が日本語モードの半角文字入力モードの画面です．

ここで 全角 を押して，全角文字入力モードにしてください．

```
N88-日本語BASIC**** Version 2.0
Copyright (C) 1981 Microsoft Corporation
Copyright (C) 1985,1986 日本電気株式会社
使用可能メモリ容量：*****バイト
Ok



全　角   ローマ字   変　換   無 変 換   重 変 換 [ローマ]
```

一番下の行に表示されているファンクションキーの内容が変わり，右端に"[ローマ]"と表示されます（ファンクションキーの内容と"[ローマ]"の意味については後述します）．

補 足　CONSOLE,,0 でファンクションキー表示が消えますが，[全角]を押すと，ファンクションキー表示が現れます．

再び [全角]を押して半角文字モードにすると，ファンクションキー表示が消えます．

注意　ファンクションキー表示を消し，カーソルを最下行に置き[全角]を押しても，全角文字入力モードにはなりません．

5 入力の手順

実際に下のようなプログラムを入力しながら，日本語の入力の手順を説明していきます．

```
10 REM 入力の練習
20 PRINT "カタカナとひらがなの入力"
30 DATA ａｂｃＡＢＣ１２３
40 A$＝"愛":B$＝"★":C$＝"凰"
```

まず半角文字入力モードで下のように入力します．入力の方法がわからないときは，もう1度第4章を見なおしてください．

```
10 REM□█
```

□はスペースを表します．

英字は，大文字でも小文字でもかまいません．

[全角] を押して，全角文字入力モードに切り替えます．

1. ローマ字入力とかな入力

次に，"にゅうりょくの"とキーボードから入力して，"入力の"に変換します．

ひらがなを入力するには，ローマ字入力方式とカナ入力方式の2通りがあります．

(1) ローマ字入力

英文のタイプライタに慣れている方なら，ローマ字で下のようにタイプすればよいでしょう．



↓

にゅうりょくの

ローマ字入力とひらがな入力の変換規則は，資料4をご覧ください．

(2) カナ入力

本機のキーには，それぞれカタカナが割り当てられていて，キートップにひらがなで記されています．カナの入力に慣れている方は，カナ入力方式を選んでください．

[かな]をロックすると，画面の右下の"[ローマ]"が"[か　な]"に変わって，カナ入力の準備ができていることを示します．

```
全　角　ローマ字　変　換　無変換　重変換 [か　な]
```

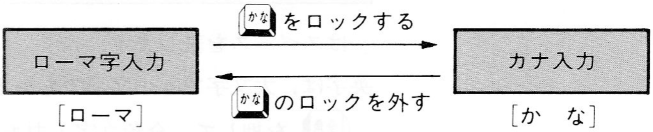
[かな]がロックされた状態で，下のようにタイプします．



↓

にゅうりょくの

ローマ字入力とカナ入力は，[かな]で切り替えます．



2. ひらがなを漢字に変換

画面には下のように，今入力した文字がリバース(白と黒が反転して)表示されています．

```
10 REM にゅうりょくの
```

リバースになっている部分は，入力の途中であり，これから変換される部分であることを示します．

ここで[変換]を押すと，ひらがなは漢字に変換されます．

```
10 REM 入力の
```

希望の漢字に変換できました．このとき，漢字に変換された"入力の"の部分はアンダーラインが付いています．

"にゅうりょくの"に対応する漢字混じりの語は"入力の"しかないので，ここでは［変換］を1度だけ押すことによって，希望の語に変換されます．

同じ読みを持つ語がいくつかあるときは，［変換］を1度押しただけでは希望の語に変換されない場合があります．もし，希望の漢字に変換されなかった場合はもう1度［変換］を押します．これによって，同じ読みを持つ別の漢字候補が現れます．

このように，［変換］を押しながら漢字候補を選んでいきます．

次の"練習"を入力するために，ローマ字入力の場合では［R す］，カナ入力の場合は［; + れ］を押すと，"入力の"の変換が終わった(決定された)とみなされてノーマル表示になります(アンダーラインやリバース表示はなくなります)．

```
10 REM 入力のれ
```

同じようにして，"練習"を入力してみましょう．

ローマ字入力では(ア)のように入力し，カナ入力では(イ)のようにキー入力します．

(ア) ［R す］［E い］［N み］［S と］［H く］［U な］［U な］　［変換］

(イ) ［; + れ］［Y ん］［D し］［シフト］+［8 (ゆ］［4 $ う］　［変換］

```
10 REM 入力の練習
```

行末で［決定］を押すと，"練習"が決定されます．［↵］を押せば，10行の入力が終了します．

［全角］を押して半角文字入力モードに切り替え，下のように入力します．

```
20 PRINT "
```

3. ひらがなをカタカナに変換する

[全角] を押して全角文字入力モードに切り替えてから，下のように入力します．

```
20 PRINT "かたかな
```

[f·10]（カタカナ）を押すと，カタカナに変換できます．

```
20 PRINT "カタカナ
```

4. ひらがなのままにしておく

続いて，"とひらがなの"と入力してから [f·4]（無変換）を押します．

```
20 PRINT "カタカナとひらがなの
```

[f·4] を押す前と何も変化がないように見えますが，これに続けて"にゅうりょく"と入力されると，下のようにリバース表示は [f·4] 以降に入力された部分だけになります．

```
20 PRINT "カタカナとひらがなのにゅうりょく
```

[変換] そして [全角] と押し，行末の " を入力します．
[⏎] を押せば，20行は完成です．

```
20 PRINT "カタカナとひらがなの入力"
```

30行目は，まず半角文字で下のように入力します．

```
30 DATA █
```

5. 英数字を全角で入力

[全角]を押して，全角文字入力モードにします．

[かな]がロックされていたら，ロックを外してください．画面の右下端には"[ローマ]"と表示されていますね．

```
 全　角   ローマ字   変　換   無 変 換   重 変 換 [ローマ]
```

ここで[f・2]（**ローマ字**）を押すと，右下の"[ローマ]"が"[英数字]"に変わります．

```
 全　角   ローマ字   変　換   無 変 換   重 変 換 [英数字]
```

[f・2]を何度か押してみると，"[ローマ]"と"[英数字]"が交互に切り替わります．

つまり[f・2]（**ローマ字**）は，英字をそのまま全角文字で入力するか（"[英数字]"），ローマ字入力としてひらがなに変換しながら入力するか（"[ローマ]"）の切り替えをします．

ただし，[f・2]（**ローマ字**）がこのような働きをするのは[かな]がロックされていない場合に限ります．

[f・2]

[ローマ] ←――→ [英数字]

"[英数字]"の状態でキー入力を行うと，全角の英数字の入力ができます．半角文字を入力するのと同じ要領です．

下のように入力してから [↵] を2回押せば，30行も完成です．

```
30 DATA ａｂｃＡＢＣ１２３█
```

```
30 DATA ａｂｃＡＢＣ１２３
```

[全角] を押して，半角文字で下のように入力しましょう．

```
40 A$="█
```

6. 1文字ずつ漢字を入力

同じ読みの漢字がたくさんあるとき，その中の1つを [変換] を何度も押しながら探すのは大変です．このようなときは [f･6] (単漢字)を使います．

[全角] を押して，全角文字で"あい"と入力してから [f･6] (単漢字)を押します．

```
1:哀 2:愛 3:挨 4:姶 5:逢                     01/05  [単漢字]
```

"あい"という読みがなの漢字が，画面の最下行に表示されます． [→] または [　　] を押すごとにリバース表示が右へ移動していきますから，"愛"のところまで移動し， [↵] を押します．

[全角] を押して半角文字入力モードに切り替え，続けて入力します(下線の部分)．

```
40 A$="愛":B$="█
```

7. 特殊文字の入力

[全角]を押して，全角文字入力モードにしておきます．

記号，ギリシャ文字，ロシア文字などは[f・7](非漢字)を使って入力します．

[f・7](非漢字)を押すと，画面の最下行に候補文字が10個ずつ表示されます．[↑]を押しながら目的の文字を探し，[→]または[　　]で目的の"★"を指して[↵]を押します．

```
1:¢  2:£  3:%  4:#  5:&  6:*  7:@  8:§  9:☆  0:★          [非漢字]
```

[全角]を押して半角文字入力モードに切り替え，下線部分を続けて入力します．

```
40 A$="愛":B$="★":C$="█
```

8. コード入力

[全角]を押して，全角文字入力モードにしておきます．

全角文字はJISコード，またはシフトJISコードで入力することができます．入力したい文字を日本語コード表(「リファレンスマニュアル」資料8参照)で探し，そのJISコードまたはシフトJISコードを調べます．ここでは"凰"を入力しますから，JISコードは16進数で5160H，シフトJISコードでは16進数で9980Hです．[f・9]（コード）を押してから，"凰"を入力したい位置に5160または9980を入力します．4桁目を入力し終わると，文字が現れます．

```
40 A$="愛":B$="★":C$="凰█
```

[全角]を押して半角文字入力モードにして，"(ダブルクォーテーションマーク)を入力し，[↵]を押せばプログラムのでき上がりです．

最後にLISTをとってみましょう．

```
list
10 REM 入力の練習
20 PRINT "カタカナとひらがなの入力"
30 DATA ａｂｃＡＢＣ１２３
40 A$="愛":B$="★":C$="凰"
Ok
```

もし間違っている箇所が見つかったら，第4章を参照して修正してください．

6 複数の文節／熟語の変換〈PC-8801MHの場合〉

N_{88}-日本語BASICの日本語入力機能では，複数の文節や熟語を含む文字列を一度に変換することはできません．たとえば，"あいじょう"を"愛情"に，"ひょうげん"を"表現"に変換することはできますが，"あいじょうひょうげん"を一度に"愛情表現"に変換することはできません．

このようなときは，f・5（重変換）を使って，複数の文節や熟語の先頭から1つずつ変換を行います．

ここでは，3つの例をあげて入力方法を説明します．

1．複合語の入力

"人工衛星"という複合語（2つ以上の熟語からできた語）を入力します．

"じんこうえいせい"と入力します．

```
じんこうえいせい
```

ここで 変換 を押してみます．変換はできません．これは，辞書に複合語が入っていないからです．このようなときは，f・5（重変換）を使います．

f・5（重変換）を押すと，下のように"じんこう"の部分だけが変換されます．

```
人口えいせい
```

このように，f・5（重変換）は，

(1) 複数の文節や熟語から成る語の先頭の語だけを変換する

という働きをします．

アンダーラインの付いている部分をさらに変換するには，変換 を使います．変換 を何度か押して希望の語を見つけてください．

```
人工えいせい
```

アンダーラインの付いている部分が現在変換の対象になっています．ここで f·5 (重変換)を押すと，下のようになります．

```
人工衛生
```

このように f·5 (重変換)は，

(2) アンダーラインの部分を決定する

(3) リバース表示の部分を変換する

という働きをします．

アンダーラインの部分をさらに変換するには 変換 を押します．

```
人工衛星
```

次に続く文字を入力することによって，"衛星"が決定されます．

2. 複文節の入力

"何らかの方法で"という，2つの文節から成ることばを入力します．"なんらかのほうほうで"と入力します．

```
なんらかのほうほうで
```

f·5 (重変換)を押すと，下のように"何らか"の部分が対象になります．

何らかのほうほうで

ここで f・5 を押してしまうと，"のほうほうで"の部分が変換の対象になってしまいます．"の"の部分をひらがなのままにしておくには，シフト + → を使います．

シフト + → を押すと，下のようになります．

何らかのほうほうで

このように シフト + → は，

(1)　アンダーラインの部分を決定する．

(2)　リバース表示になっている部分を1文字ずつ決定する．

という2つの働きをします．

誤って シフト + → を押し過ぎてしまい，決定した部分が長くなってしまった場合は，シフト + ← で1文字ずつもとに戻すことができます．たとえば，下のようになってしまった状態で シフト + ← を2回押せば，もとに戻ります．

何らかのほうほうで

↓ シフト + ←

何らかのほうほうで

＊この場合は 変換 を使うこともできますが，複数の文節や熟語から成る語の変換は，原則として f・5 (重変換)を使うようにしてください．

変換 で，残りのリバース表示の部分を変換します．

何らかの方法で

次に続く文字を入力することによって，"方法で"が決定されます．

3. ひらがなの文節と漢字を含む文節

"そんな馬鹿な"という，ひらがな／漢字混じりの複文節を入力します．

"そんなばかな"と入力します．

そんなばかな

これをいきなり［変換］や［f・5］で変換すると，文字列の頭の部分から変換してしまいます．たとえば，下のようになってしまいます．

損なばかな

"そんな"の部分はひらがなのままにしておきたいわけですから，［ESC］を押してもとに戻します．

そんなばかな

［シフト］＋［→］を3回押して，"そんな"をひらがなに決定します．

そんなばかな

［変換］を押して，残りの部分を変換します．

そんな<u>馬鹿な</u>▌

次に続く文字を入力することによって，"馬鹿な"が決定されます．

7　複数の文節/熟語の変換〈PC-8801FHの場合〉

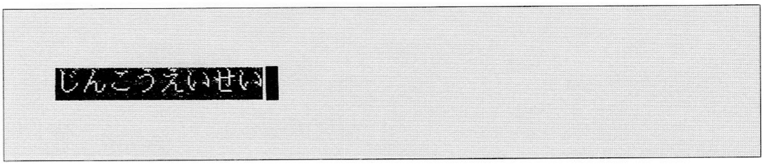
N$_{88}$-日本語BASICの日本語入力機能では，複数の文節や熟語を含む文字列を一度に変換することはできません．たとえば，"あいじょう"を"愛情"に，"ひょうげん"を"表現"に変換することはできますが，"あいじょうひょうげん"を一度に"愛情表現"に変換することはできません．

文節や熟語の区切りがわかりにくくて，文字列の一部しか漢字に変換できなかったとき，[f・5]（重変換）を使って続けて残りの部分の変換をすることができます．ここでは，3つの例をあげて入力方法を説明します．

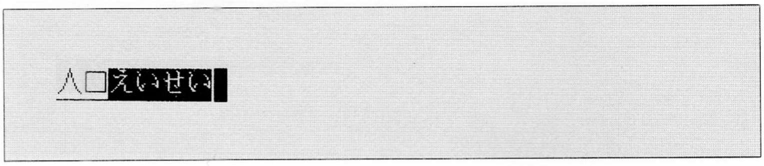
1．複合語の入力

"人工衛星"という複合語（2つ以上の熟語からできた語）を入力します．

"じんこうえいせい"と入力します．



[f・5]（重変換）を押すと，下のように"じんこう"の部分だけが変換されます．



このように，[f・5]（重変換）は，

(1)　複合語の先頭の語だけを変換する

という働きをします（[変換]でも同じ働きをします）．

アンダーラインの付いている部分をさらに変換するには[変換]を使います．[変換]を何度か押して，希望の語を見つけてください．

アンダーラインの付いている部分が，現在変換の対象になっています．

ここで f·5 （重変換）を押すと，下のようになります．

人工衛生

このように f·5 （重変換）は，

(1)　アンダーラインの部分を決定する

(2)　リバース表示の部分を変換する

という2つの働きをします．

アンダーラインの部分をさらに変換するには 変換 を押します．

人工衛星

次に続く文字を入力することによって，"衛星"が決定されます．

2. 複文節の入力

"何らかの方法で"という2つの文節から成ることばを入力します．

"なんらかのほうほうで"と入力します．

なんらかのほうほうで

変換 を押すと，下のように"何ら"の部分が変換の対象になります．

何らかのほうほうで

ここで f·5 を押してしまうと，"かのほうほうで"の部分が変換の対象になってしまいます．"かの"の部分をひらがなのままにしておくには，シフト ＋ → を使います．

[シフト]＋[→]を2回押すと，下のようになります．

```
何らかのほうほうで
```

このように[シフト]＋[→]は，

(1) アンダーラインの部分を決定する

(2) リバース表示になっている部分を1文字ずつ決定する

という2つの働きをします．

誤って[シフト]＋[→]を押し過ぎてしまい，決定した部分が長くなってしまった場合は，[シフト]＋[←]で1文字ずつもとに戻すことができます．たとえば，下のようになってしまった状態で[シフト]＋[←]を2回押せば，もとに戻ります．

```
何らかのほうほうで
```

↓ [シフト]＋[←]

```
何らかのほうほうで
```

＊この場合は[変換]を使うこともできますが，複数の文節や熟語から成る語の変換は，原則として[f･5]（重変換）を使うようにしてください．

[変換]で，残りのリバース表示の部分を変換します．

```
何らかの方法で
```

次に続く文字を入力することによって，"方法で"が決定されます．

3. ひらがなの文節と漢字を含む文節

"そんな馬鹿な"という，ひらがな／漢字混じりの複文節を"そんなばかな"と入力します．

```
そんなばかな
```

これをいきなり［変換］で変換すると，文字列の頭の部分から変換してしまいます．たとえば，下のようになってしまいます．

```
損なばかな
```

"そんな"の部分はひらがなのままにしておきたいわけですから，［ESC］を押してもとに戻します．

```
そんなばかな
```

［シフト］＋［→］を3回押して，"そんな"をひらがなに決定します．

```
そんなばかな
```

［変換］を押して，残りの部分を変換します．

```
そんな馬鹿な
```

次に続く文字を入力することによって，"馬鹿な"が決定されます．

8 各キーの使い方

日本語文字の基本的な入力方法は，前節まででひととおり説明してきました．

ここでは，日本語の入力に慣れた方が参照できるように，各キーごとにその使い方をまとめます．

キー	機能
[全角] ([f·1])	全角文字入力モードと半角文字入力モードとの切り替えを行います． 全角文字入力モードでは，[f·2] ～ [f·10] がこの表に示されている内容になります．
[f·2] ([ローマ字] or [ローマ]) 画面の右下に，ローマ字入力モードのときは[ローマ]，英数字入力モードのときは[英数字]と表示されます．	ローマ字入力モードと英数字入力モードの切り替えを行います． ローマ字入力モードでは，アルファベット(A～Z)を入力すると，ローマ字→ひらがな変換規則に従ったひらがなに変換されます． 英数字入力モードでは，アルファベットはそのまま入力されます． [かな] がロックされているときは，常にカナ入力になりますので，このキーは意味を持ちません．
[変換] ([f·3])	リバース表示されている，またはアンダーラインのついた範囲の文字列を文節単位(PC-8801MHの場合)，熟語単位(PC-8801FHの場合)に漢字に変換します． このキーを繰り返し押すことにより，次々に漢字候補が現れます． 候補は次のいずれかの動作によって決定されます． (1) 次の文字を入力する． (2) [決定] または [全角] を押す． 単漢字変換([f·6])の途中からでも，このキーを押すことによって，この変換に切り替えることができます．
[無変換] ([f·4])	入力された文字列を，変換せずにそのまま決定します．また，変換中の文字列を入力したときの状態に戻します．
[f·5] ([重変換])　PC-8801MH	複合語や複文節の変換に使用します． ① 複数の文節や熟語からなる語の先頭の語だけを変換します． ② 複合語や複文節の途中までが変換の対象になっている(アンダーラインが付いている)とき，その部分を決定し，それに続く語を変換します．
[f·5] ([重変換])　PC-8801FH	複合語の変換を行う場合に使用します． 複合語の途中までが変換対象になっているとき，[f·5] を押すことによって，変換中の部分を決定し，それに続く語を変換します．

<table>
<tr><th>キー</th><th>機能</th></tr>
<tr><td>f・6 (単 漢 字)
単漢字の選択中は，画面右下に[単漢字]と表示されます．</td><td>変換の対象の読みを持つ，1文字の漢字に変換します．
また，次のような読みを漢字以外の文字に変換します．
(1)　「ぎりしあ」→ f・6 →ギリシャ文字
(2)　「ろしあ」　→ f・6 →ロシア文字
(3)　「かな」　　→ f・6 →特殊かな(ゎゐゑヮヰヱヵヶ)
(4)　「がいじ」　→ f・6 →外字
f・6 を押すと，画面の最下行に，10文字ずつ候補文字が表示されます（40文字/行のときは5文字ずつ）．11文字以上の候補がある場合は，↑，↓ によって次の候補群，前の候補群を表示させます．[単漢字]の左の数字によって，全部で何文字の候補があり，その中の何番目の候補が表示されているかがわかります．たとえば次のように表示されているときは，49文字のうちの20番目の候補を指しているということになります．

1:始　2:姉　3:姿　4:子　5:屍　6:市　7:師　8:志　9:思　0:指　20/49　[単漢字]

候補は次の方法のいずれかで決定します．
(1)　0～9の数字を入力する．
(2)　→，← で文字を選んで ⏎ を押す．</td></tr>
<tr><td>f・7 (非 漢 字)
非漢字の選択中は，画面右下に[非漢字]と表示されます．</td><td>この機能によって，次の文字を入力します．
(1)　記号(JISコード 2121H～222EH)
(2)　ギリシャ文字
(3)　ロシア文字
(4)　特殊かな(ゎゐゑヮヰヱヵヶ)
(5)　外字
f・7 を押すと，画面の最下行に候補文字が表示されます．
ただし，未決定の文字があるときは非漢字の入力をすることはできません．

1:■　2:、　3:。　4:，　5:．　6:・　7:：　8:；　9:？　0:！　[非漢字]

候補の選択方法は，単漢字と同じです．</td></tr>
<tr><td>f・8 (前 候 補)</td><td>変換 によって，候補文字列を選択している途中で，f・8 を押すことによって，1つ前の候補に戻すことができます．
同じ読みを持つ候補がたくさんあるときに誤って 変換 を押しすぎたときに，f・8 で前の候補に戻します．</td></tr>
</table>

キ　ー	機　能
f·9 （コード） コードの入力中は画面右下に[コード]と表示されます.	JISコードまたは，シフトJISコードを16進数で指定することによって，全角文字を入力します. f·9 を押した後，4桁目の16進数を入力します(3桁目までで，入力を誤ったときは 削除 によって修正することができます). 4桁目の入力と同時に，そのコードで表される全角文字が表示されます. ただし，未決定の文字があるときはコードの入力はできません.
f·10 （カタカナ or カ　ナ）	リバース表示されている，変換対象文字列をカタカナに変換します. カタカナに変換したあとで，f·4 によって，ひらがなに戻したり，変換 によって，漢字へ変換を続けることができます.
決定　未決定の文字がある場合，そのまま決定します.	未決定の文字がある場合は，決定 によって決定します.
↵	未決定の文字がある場合は，↵ によって決定します. 未決定の文字がない場合は，改行を行います.
ESC	ESC は以下の2つの働きをします. (1)　変換，f·5（重変換），f·10（カタカナ）で表示される候補(リバース表示，または下線つきで表示されているもの)を，入力直後の状態(ひらがな)に戻します. (2)　単漢字入力(f·6)，非漢字入力(f·7)，およびコード入力(f·9)の途中で，入力を中断します.
かな ロックされていると，画面の右下に[か　な]と表示されます.	かな をロックすると，カナ入力モードになり，各キーに記されているカタカナに相当するひらがなが入力できます. ロックを外すと，ローマ字でのひらがな入力または，英数字の入力モードになります.
シフト ＋ →	複分節の語の変換の際，現在変換の対象になっている語を決定し，それに続く文字を1文字ずつ決定します. または，シフト ＋ ← によって未決定にした文字列の左端から1文字ずつ決定します.
シフト ＋ ←	シフト ＋ → などによってひらがな，カタカナ，英数字に決定した部分を1文字ずつ，もとの未決定の状態に戻します.
（スペース）	全角のスペースが入力されます. スペースの左隣の文字列が変換の途中だった場合は，スペースの入力によって決定されます.
← →	変換中の文字列を決定し，カーソルを左右に移動します.

9 注意事項

日本語モードでの入力の際には，次の点にご注意ください．

- 変換途中の文字列(リバース表示，または下線付きになっている文字列)があるときは，[↑]，[↓]によってカーソルを上下に動かすことはできません．
- 単漢字と非漢字の文字候補の語順には，学習機能がありません．

第6章

日本語処理
(N88-日本語BASIC)

本機は，JIS第1，2水準の日本語文字を格納したROMを標準で備えています．
また，プログラミング言語としてN88-BASICに，日本語処理機能を付加したN88-日本語BASICを用意し，強力な日本語処理機能を持っています．
この章では，本機の持つ日本語処理機能をより有効にお使いいただくための説明をします．

N88-日本語BASICを使用するためには，N88-日本語BASICシステムディスクが必要です．

1 日本語処理とは

日本語処理機能を使用することにより，日本語文字列を，英数カナと同じようにPC-PR201系プリンタのような漢字出力可能なプリンタに印字することができます．

日本語処理機能を整理すると，次のようになります．

1．日本語入力

文節変換入力（PC-8801FHは熟語変換入力），表示選択入力などの入力機能を使用し，キーボードからの日本語入力が可能です．

2．文字列定数としての日本語文字列定義

プログラムテキストの中での文字列定数として日本語文字列を定義するもので，英数カナと同じように" "（ダブルクォーテーション）でくくって定義します．

日本語文字列としては，定数として定義する場合と，文字列変数を引用する場合とがあります．

たとえば，「日本語」という文字列をPRINT命令の中で定数として定義する場合，次のように入力してください．

```
print "日本語"
```

```
PRINT "日本語"
日本語
Ok
```

"日本語"という文字が画面に表れましたね．

あるいは「日本語」を，PRINT命令の中で文字列変数を引用する場合，次のように入力してください．前と同じように"日本語"が画面に表れます．

```
a$＝"日本語"
print a$
```

```
a$="日本語"
print a$
日本語
Ok
```

3. 日本語文字列の印刷

日本語文字列をPC-PR201系の漢字出力可能なプリンタに出力すれば，日本語文字を印刷することができます(PC-PR201系プリンタを使用すると明朝体の印字が可能です).

lprint "日本語" ⏎

4. 日本語文字列の操作

日本語文字列を操作するために種々の関数が用意されており，それらの関数を使用することにより，たとえば，文字列中の全角文字を1バイト系英数カナ文字に変換したり，文字列に含まれる文字数を数えたりできます.

一例として，次のとおりに入力してください．画面に指定した文字列の文字数が"7"と表示されます.

? KLEN("NEC日本電気") ⏎

```
? KLEN("ＮＥＣ日本電気")
 7
Ok
```

5. 日本語文字列のファイル入出力

日本語文字列も1バイト系英数カナ文字と同じようにファイルに出力することができます．下のプログラムは，日本語文字列をディスクファイル(シーケンシャルファイル)に出力し，すぐに読みなおします．下に示すように入力してください)．

```
10 open "DATA" for output as #1 ↵
20 print #1, "日本語" ↵
30 close ↵
40 open "DATA" for input as #1 ↵
50 input #1, a$ ↵
60 print a$ ↵
70 close ↵
RUN ↵
```

シーケンシャルファイルについては，第14章を参照してください．

2 日本語文字(全角文字)

1. 半角文字と全角文字

BASICでPRINT文，INPUT文，REM文や文字定数，文字変数(変数名に$が付くもの)，STRING$，MID$などの関数によって様々な文字列を扱います．

N88-BASICなどの日本語処理機能を持たないBASICでは，文字列として本書資料2の「キャラクタコード」にあげられている256種類の1バイト系英数カナ文字(これを半角文字といいます)だけを扱います．

これに対してN88-日本語BASICでは，半角文字に加えて日本語コードにあげられている6349種類の漢字と453種類の漢字以外の文字(これを全角文字といいます)を扱うことができます．

なお，日本語コードについては「リファレンスマニュアル」を参照してください．

2. シフトJISコードについて

N88-日本語BASICは，全角文字を"シフトJISコード"と呼ばれるコードで扱います．

日本語文字は，一般的にはJISの規格で定められている"JISコード"と呼ばれるコードで扱われていますが，N88-日本語BASICでは，内部で処理しやすいシフトJISコードを採用しています．

したがって，外部の機器との間で日本語データをやりとりする場合や，N88-日本語BASICで作ったファイルを他のコンピュータが読み取る場合などは，シフトJISコードをJISコードに変換する必要があります．

シフトJISコードとJISコードの変換のサブルーチンの例をあげておきます．

シフトJISコード→JISコード変換サブルーチン

JIS$にJISコードを入れておくと，SHIFTJIS$にシフトJISコードを入れて返すサブルーチンです．

```
100 'シフトJIS→JIS　コード変換サブルーチン
110 SJH=VAL("&H"+LEFT$(SHIFTJIS$,2))
120 SJL=VAL("&H"+RIGHT$(SHIFTJIS$,2))
130 IF SJL<=&H9E THEN GOSUB *LESSER9EH ELSE GOSUB *GREATER9EH
140 JIS$=HEX$(JH)+HEX$(JL)
150 RETURN
160 *LESSER9EH
170 IF SJH<=&H9F THEN JH=(SJH-&H71)*2+1 ELSE JH=(SJH-&HB1)*2+1
180 IF SJL>=&H80 THEN JL=SJL-&H1F-1 ELSE JL=SJL-&H1F
190 RETURN
200 *GREATER9EH
210 IF SJH<=&H9F THEN JH=(SJH-&H70)*2 ELSE JH=(SJH-&HB0)*2
220 JL=SJL-&H7E
230 RETURN
```

JISコード→シフトJISコード変換サブルーチン

SHIFTJIS$にシフトJISコードを入れておくと，JIS$にJISコードを入れて返すサブルーチンです．

```
100 'JIS→シフトJIS　コード変換サブルーチン
110 JH=VAL("&H"+LEFT$(JIS$,2))                    ' JH:JISコードの1バイトめ
120 JL=VAL("&H"+RIGHT$(JIS$,2))                   ' JL:JISコードの1バイトめ
130 IF JH<=&H5E THEN SJH=INT((JH-1)/2)+&H71
                ELSE SJH=INT((JH-1)/2)+&HB1       'SJH:シフトJISコードの1バイトめ
140 IF JH MOD 2=0 THEN SJL=JL+&H7E:GOTO 160
                  ELSE SJL=JL+&H1F                'SJL:シフトJISコードの2バイトめ
150 IF SJL>=&H7F THEN SJL=SJL+1
160 SHIFTJIS$=HEX$(SJH)+HEX$(SJL)                 'SHIFTJIS$:シフトJISコードを
                                                   16進数の文字列で返す
170 RETURN
```

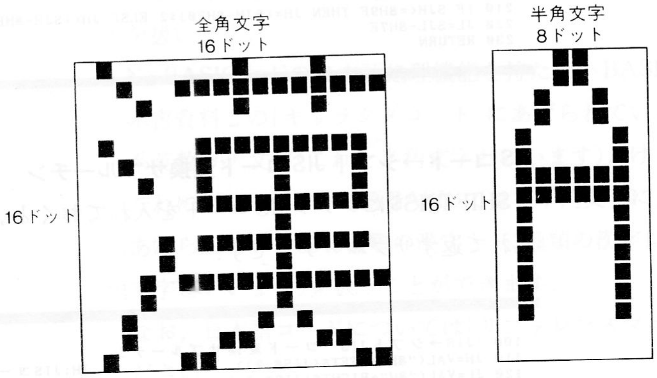
3. 全角文字が使える部分

全角文字は，次のような部分に使うことができます．

(1) 文字列の内容
(2) 文字定数(" "で囲まれた文字列)
(3) REM文の中の注釈
(4) DATA文の中の文字定数
(5) ファイル名
(6) INPUT(WAIT)，LINEINPUT(WAIT)文のプロンプト文
(7) ファンクションキーの内容

3 画面構成

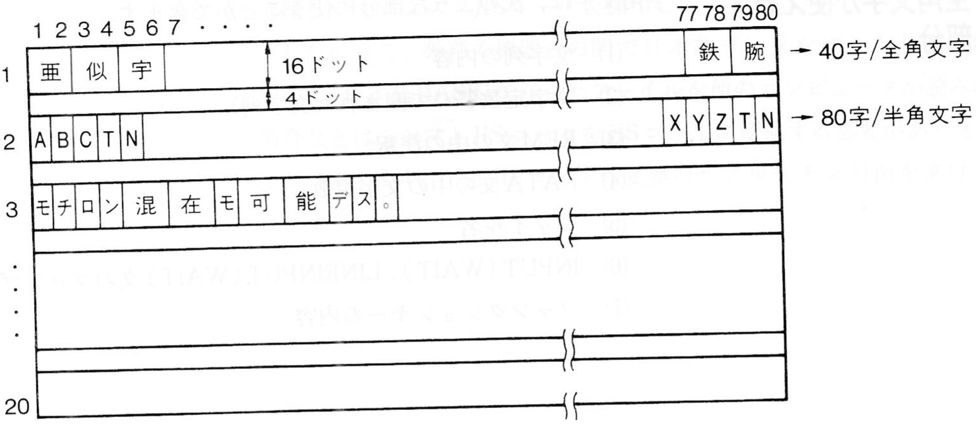
全角文字は漢字などを表示するので，下図に示すように半角文字とはドットパターンが違います．



したがって，全角文字は半角文字２文字分のスペースを必要とします．つまり，１行をすべて全角文字で埋めた場合は，半角文字の半分しか表示できません．

もちろん，下図に示すように全角文字と半角文字を混在させることもできます．

WIDTH 80, 20 : SCREEN 4の場合



この例の場合，１画面には40×20＝800個の全角文字が表示できることになります．

WIDTH文の第１パラメータ（１行の中の文字数）は，半角文字で数えた値になりますから注意してください．

WIDTH文の設定と，そのときに表示できる全角文字とは，次ページ

の表に示すような関係になります．

SCREEN 3の場合

WIDTH文の指定	表示できる全角文字の数
WIDTH 80, 12	40文字×12行＝480文字
WIDTH 80, 10*	40文字×10行＝400文字
WIDTH 40, 12	20文字×12行＝240文字
WIDTH 40, 10*	20文字×10行＝200文字

SCREEN 4の場合

WIDTH文の指定	表示できる全角文字の数
WIDTH 80, 25	40文字×25行＝1000文字
WIDTH 80, 20*	40文字×20行＝ 800文字
WIDTH 40, 25	20文字×25行＝ 500文字
WIDTH 40, 20*	20文字×20行＝ 400文字

*　SCREEN文を実行すると，WIDTH文の第２パラメータ（行数/画面）は，自動的に10または20に設定されます．

4 日本語処理用関数

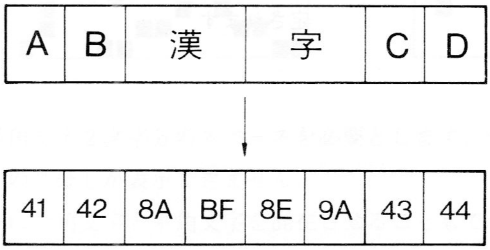
日本語処理用の関数として，KPOS，KLENなどがあります．

これらの関数については「リファレンスマニュアル」を参照してください．

BASIC中の文字および文字列を操作する関数やステートメントは，すべて半角文字と全角文字を混在して扱えるようになっています．

このため，INSTR関数，MID$関数，RIGHT$関数，LEFT$関数が期待どおりの動作をしないことがあります．これらの関数は，全角文字，半角文字を区別して扱わず，全角文字は半角文字が２個並んでいるものとして動作をするためです．

たとえば"AB漢字CD"という文字列は，下図に示すように８バイトの半角文字とみなされます．



ここでは，MID$関数とINSTR関数を例にあげて，文字列の扱いに関する留意点を説明していきます．

1．MID$関数

下に示すとおりに入力してください．

```
print mid$("AB海老",3,2)
```

"AB海老"という文字列の３バイト目から２バイトの長さの文字列を切り取って表示させています．"海"は１文字で２バイトの長さがありますから，正しい結果が得られたわけです．

今度は下に示すように入力し，実行してください．

```
print mid$("AB海老",4,2)
```

おかしな結果になりましたね．

"海老"をシフトJISコードで表すと，8A43H 9856Hとなります．今の場合，"AB海老"の４バイト目は，"海"の後ろ半分の43Hです．そこから２バイト分を切り取るということは"老"の前半分である98Hまでを切

り取って1つの文字のデータとして表示するということになります(98Hだけでは文字のデータとして定義されていないので，実際には半角文字のCが表示されます).

このように文字列の切り分け方に注意しないと，場合によって，おかしな文字が表示されてしまいます.

全角文字を含む文字列を操作する場合は，全角文字の半分だけを切り取らず，1文字分を2バイト単位で組にして扱うようにしてください.

全角文字と半角文字との混在する文字列で，半角文字と同じように全角文字も1文字ずつ扱いたい場合には，KLEN関数，KPOS関数と組み合わせてプログラムを組んでください.

LEFT$, RIGHT$, MID$ステートメントも同様の注意が必要です.

2. INSTR関数

同じ文字"S"を探す場合でも，うまくいく場合とうまくいかない場合があります．まずは，下図に示すように入力し実行してください.

うまくいく場合

```
PRINT INSTR("日本語BASIC","S")
 9
Ok
```

うまくいかない場合

```
PRINT INSTR("鬼が島BASIC","S")
 2
Ok
```

どうですか？ 結果が違いますね.

これは，BASICが全角文字をシフトJISコードにして処理するために起こるものです.

上図でうまくいかない場合では，文字列の中の"鬼"のシフトJISコードは8B53Hで，このうち下位バイトの53Hが"S"のキャラクタコードと同じため，BASICは，ここで発見したとみなしてしまうからです．これは，調べたい文字が半角文字1文字のときのみに起こり，半角文字が2文字以上，および全角文字のときには起こりません.

この問題を解決するサブルーチンを含むプログラムが次ページに示してあります.

```
100 'INSTR 補正プログラム
110 '
120 X$="鬼が島BASIC"
130 Y$="S"
140 N=1
150 '
160 GOSUB *ADVANCED.INSTR
170 PRINT Y$;" は、";X$;" の";NO;"バイトめにあります。"
180 END
190 '
200 *ADVANCED.INSTR
210 IF LEN(Y$)=1 AND ASC(Y$)<256 THEN 250
220 ' Y$が、2バイト以上のとき
230 NO=INSTR(N,X$,Y$)
240 GOTO 330
250 ' Y$が、半角文字1文字のとき
260 Y=ASC(Y$):I=N
270 C=KPOS(X$,I)
280 IF C=0 THEN 330
290 W$=MID$(X$,C,1)
300 IF ASC(W$)=Y THEN NO=I
310 IF NO=0 THEN I=I+1:GOTO 270
320 NO=KPOS(X$,NO)
330 RETURN
```

このプログラムは，半角文字が1文字のときだけ別のサブルーチンで判断しているので，正しい結果が得られるわけです．

このプログラムでは，120行〜140行でX$に調べる文字列を代入し，Y$に調べたい文字列を代入します．そして，何バイト目から調べるかをNに代入します．それから，200行から始まるサブルーチンを実行します．

サブルーチンの中では210行で，Y$が1文字で半角文字の場合とそうでない場合の区別をします．そして，1文字で半角文字の場合は250行へ実行が移り，ここで正しい判断をします．また，そうでない場合はINSTR関数で処理できるので，220行へ実行が移ります．どちらもNOという変数に答えを代入して，330行のRETURN文でサブルーチンを終わります．その後170行に帰り，NOを表示してプログラムを終了します．

これにより，INSTR関数は正確に動作できるようになります．したがって，INSTR関数で半角文字1文字の判断をするときは，このプログラムの250行から310行を参考にして，プログラムを作成してください．

さて，INSTR関数は，調べたい文字がその文字列の左から何バイト目にあるかを返すものですが，何文字目にあるかを得たい場合もあります．特に全角文字を使用していると，INSTR関数では，バイト数＝文字数が成立しないので，文字列操作が厄介なものになります．そこで，拡張INSTR関数とでもいうプログラム(サブルーチン)を作成しました．次ページに示したものがそうです．

```
100 ' 拡張INSTR関数プログラム
110 X$="鬼が島BASIC"
120 Y$="S"
130 N=1
140 '
150 GOSUB *ENHANCED.INSTR
160 PRINT Y$;" は、";X$;" の";NO;"文字めにあります。"
170 END
180 *ENHANCED.INSTR
190 IF LEN(Y$)=1 AND ASC(Y$)<256 THEN 270
200 '
210 ' Y$が、２バイト以上のとき
220 NO=INSTR(N,X$,Y$):IF NO=0 THEN 340
230 C$=LEFT$(X$,NO-1)
240 C=LEN(C$)-KLEM(C$):NO=NO-C
250 GOTO 340
260 '
270 ' Y$が、半角文字１文字のとき
280 Y=ASC(Y$):I=N
290 C=KPOS(X$,I)
300 IF C=0 THEN 340
310 W$=MID$(X$,C,1)
320 IF ASC(W$)=Y THEN NO=I
330 IF NO=0 THEN I=I+1:GOTO 290
340 RETURN
```

これにより，NOという変数に文字数(半角文字と全角文字混在のキャラクタ数)を返すことができます．プログラムの内容については，ほとんどINSTR関数のプログラムと同じなので，そちらを参照してください．

5　日本語文字のプリンタへの出力

N₈₈-日本語BASICは，漢字などの全角文字を出力できる日本語プリンタをサポートしています．LPRINT文やLLIST文を使って，全角文字を含む文字列やプログラムを日本語プリンタに印字することができます．また，COPY文や，[コントロール]＋[画面コピー]で，プリンタの文字フォントを使った画面コピーも可能となっています．

ここでは，主なプリンタのうち，N₈₈-日本語BASICに適したプリンタ，使用するときに注意が必要なプリンタなどについて解説します．

1. N₈₈-日本語BASICに適したプリンタ

下図に示したプリンタは，N₈₈-日本語BASICのすべての機能がサポートされています．

24ドット日本語プリンタ	PC-PR201F，PC-PR101F
	PC-PR201H2
	PC-PR201V
	PC-PR201TL
	PC-PR101TL
	PC-PR406
	PC-PR406M

2. 注意が必要なプリンタ

(1) 半角文字と全角文字の比が1：2にならないプリンタ

下図に示したプリンタに全角文字を含む文字列を印字した際，半角文字と全角文字の比が1：2になっていませんので，桁が揃わない場合があります．

24ドット日本語プリンタ	PC-PR104
	PC-PR405
18ドット日本語プリンタ	PC-PR202K
16ドット日本語プリンタ	PC-PR403

これを補正するためには，システムディスクに入っているユーティリティを使用してください．

システムディスクがドライブ1に入っていることを確認したうえで，

```
run "setupdt.j88" ↵
```

を実行します．

6 注意事項

1. 文字列定数の有効文字数

文字列定数は最大255桁ですが，全角文字を含む場合，全角文字の１文字が半角文字２文字に対応しているため，実際の有効文字数はそれ以下となります．

すべての全角文字から成る文字定数の最大文字数は127文字です．

2. 日本語文字列のランダムファイルへの出力

ランダムファイルに日本語文字列を含んだデータを出力する場合，全角文字１文字が英数カナ文字２文字に対応することを考えてフィールド長を決定する必要があります．

```
10 a$＝"日本語" ↵
20 lset fa$＝a$ ↵
30 put ♯1, i ↵
```

この場合，フィールドfa$のフィールドは最低６文字必要となります．

```
field ♯1, 10 as fa$, ････
```

3. ファンクションキーのコピー

N88-日本語BASICの日本語モードでCOPY 1 ↵ を行った場合，ファンクションキーの内容はコピーされません．

第7章

日本語ユーティリティ

N88-日本語BASICシステムディスクには，外字とユーザ辞書を管理したり，また，他のBASICとのファイル交換を可能にするためのユーティリティプログラムが入っています．ここでは，これらのユーティリティプログラムの使い方を説明します．

1 日本語ユーティリティの概要

この章では，N_{88}-日本語BASICシステムディスクに納められている4つのユーティリティプログラムの使い方を説明します．

4つのユーティリティプログラムのファイル名と，機能をまとめます．

ファイル名	機　　能
font. mh (PC-8801MHの場合) font. fh (PC-8801FHの場合)	外字管理 システムディスク上に書き込まれている，外字のフォント(書体)の定義，変更を行う． 外字は画面表示用(16×16ドット)と，プリンタ印字用(24×24ドット)がある．
prload. mh (PC-8801MHの場合) prload. fh (PC-8801FHの場合)	プリンタへの外字ロード 外字をプリンタに送る．
usrdic. mh (PC-8801MHの場合) usrdic. fh (PC-8801FHの場合)	ユーザ辞書管理 システムディスクに入っている，ユーザ辞書への熟語の登録，変更などを行う．
cdcnv. mh (PC-8801MHの場合) cdcnv. fh (PC-8801FHの場合)	コード変換 N_{88}-日本語BASICと他のBASICとの間でのデータ交換をするために，ファイル中の文字列の表現を変換する．

N_{88}-日本語BASICでは，必ず本体右側のミニフロッピーディスクドライブにN_{88}-日本語BASICシステムディスクをセットしておく必要があります．

これは，ディスクコード(日本語BASICを動かすもの)以外にデータ(辞書，ユーザ辞書，外字データ)がシステムディスクに書き込まれていて，N_{88}-日本語BASICの起動中に，システムディスクから頻繁にデータを読み込む必要があるからです．

この章で説明する外字管理，プリンタへの外字ロード，およびユーザ辞書管理のユーティリティでは，すべてN_{88}-日本語BASICシステムディスクが操作の対象になることにご注意ください．

この章で使われている"システムディスク"とは，N_{88}-日本語BASICシステムディスクを指しています．

2 外字管理

[font.fh(PC-8801FH), font.mh(PC-8801MH)]

N_{88}-日本語BASICでは，本機に実装されている漢字ROMに入っている日本語文字を扱う機能を持っています．それに加えて，ユーザが作った文字／記号(外字)等をN_{88}-日本語BASICシステムディスク上に書き込み，BASIC命令で扱うことができます．

N_{88}-日本語BASICのKPLOAD文は外字を定義するための命令ですが，KPLOAD文を使って，プログラム上で外字を定義するのはかなり厄介です．また，KPLOAD文では，システムディスク上の外字辞書の書き換えはできません．

しかし，この外字管理ユーティリティを使うことにより，簡単に外字を定義することができます．また，あるシステムディスク上の外字データを他のシステムディスクに転送したり，プリンタに対して外字データを送ることもできます．

外字管理ユーティリティは，外字のコードとしてJISコードを用います．JISコードの7621Hから765FHまでの63文字分の領域が，外字専用に確保されています．

N_{88}-日本語BASICでは，全角文字(外字を含めて)をシフトJISコードで表現しますから，このユーティリティで作成した外字をN_{88}-日本語BASICで扱う場合は，次の表を参照して，シフトJISコードを調べてください．

JISコード	シフトJISコード	JISコード	シフトJISコード	JISコード	シフトJISコード	JISコード	シフトJISコード
		7630	EBAE	7640	EBBE	7650	EBCE
7621	EB9F	7631	EBAF	7641	EBBF	7651	EBCF
7622	EBA0	7632	EBB0	7642	EBC0	7652	EBD0
7623	EBA1	7633	EBB1	7643	EBC1	7653	EBD1
7624	EBA2	7634	EBB2	7644	EBC2	7654	EBD2
7625	EBA3	7635	EBB3	7645	EBC3	7655	EBD3
7626	EBA4	7636	EBB4	7646	EBC4	7656	EBD4
7627	EBA5	7637	EBB5	7647	EBC5	7657	EBD5
7628	EBA6	7638	EBB6	7648	EBC6	7658	EBD6
7629	EBA7	7639	EBB7	7649	EBC7	7659	EBD7
762A	EBA8	763A	EBB8	764A	EBC8	765A	EBD8
762B	EBA9	763B	EBB9	764B	EBC9	765B	EBD9
762C	EBAA	763C	EBBA	764C	EBCA	765C	EBDA
762D	EBAB	763D	EBBB	764D	EBCB	765D	EBDB
762E	EBAC	763E	EBBC	764E	EBCC	765E	EBDC
762F	EBAD	763F	EBBD	764F	EBCD	765F	EBDD

1. 外字管理ユーティリティをスタートする

N_{88}-日本語BASICをスタートさせ，"font.～"をロードし，実行します．

注意 ユーティリティプログラムの入っているフロッピーディスクをドライブ1（8インチディスクユニットが接続されているときはドライブ3）にセットしてください．この例は，ドライブ1にシステムディスクがセットされていると想定したときのものです．

```
load "font.～" ⏎
ok
run ⏎
```

しばらくすると次のように表示されます．

```
変更したい外字の入っているシステムディスクをセットして
ドライブ番号を入力してください：█
```

メッセージに対して，システムディスクのセットされているドライブ番号を，たとえば，

```
1 ⏎
```

と入力します．

注意 8インチディスクユニットが接続されていれば，ドライブ番号は3になります．
数字は半角文字で入力してください．
以下，メニューで処理を選んだり，ドライブ番号を入力するときも同様です．

メニュー画面が表示されます．

```
<< 外 字 管 理 >>

1. 作 成 .....登録されている外字の修正
2. 転 送 .....データを別のメディアに転送


どの処理を行いますか？█
```

2. 外字を作る

メニュー画面で

1 ↵

と入力します．

実際に外字を作りながら手順を説明していきます．Ⓡというマークを作ってみましょう（Ⓡは漢字ROMに入っていませんから，外字として定義しないと表示できません）．

(1) 外字コードを定義する

外字コードの7621HにⓇを定義することにしましょう．

注意 現在7621Hに定義されている外字（買ったばかりのときは"↵"が定義されています）が更新されます．

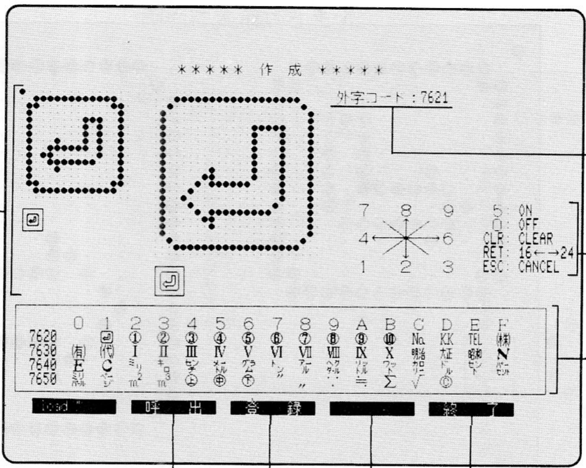
```
外字コード： █
```

に対して，

7621 ↵

と入力すると，次ページのような作成画面が表示されます．

作成画面



作成中の外字コードが表示されます.

テンキーの使い方が表示されます.

現在定義されている外字の一覧表が表示されます.
標準ディスプレイ（640×200ドット）を使っているときは表示されません.

f・5
作成を終了してメニューに戻ります.
このとき，プリンタが接続されていればプリンタに外字データを送ります.

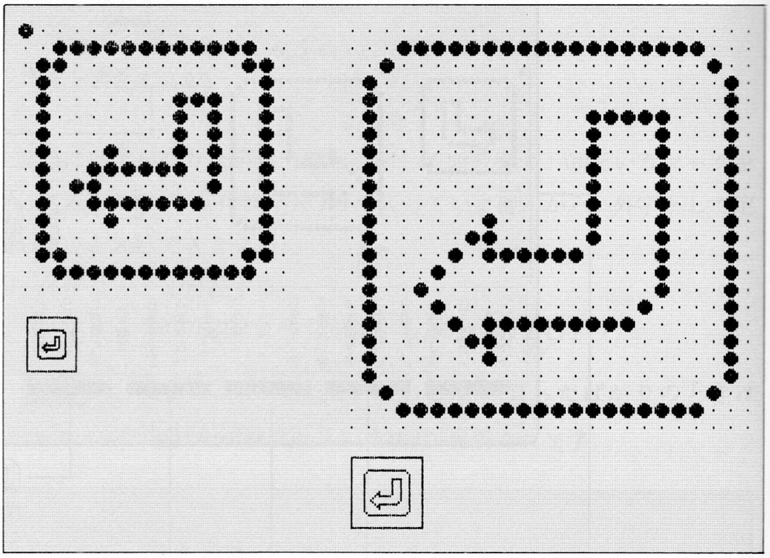
f・4
標準ディスプレイを使っているときは，**f・4** には **一覧** と設定されていて，これを押すことによって，現在定義されている外字の一覧表が表示されます.

f・3
作成中の外字をシステムディスクの外字辞書に登録します.

f・2
漢字ROMに入っている文字や，現在定義されている外字のフォントを呼び出します.

外字作成エリアです.
16×16ドットの文字は画面表示用，24×24ドットの文字は，24ピン日本語プリンタ用です.
16ピンの日本語プリンタでは，外字を印字することはできません.

画面には，現在定義されているフォントが表示されます．



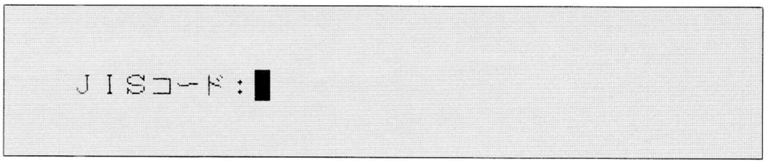
(2) 外字をデザインするとき

外字をデザインするときは，既存のフォントに似たものがあれば，それを呼び出して修正するのが便利です．

注意　既存の文字をもとにしてフォントを作るのがやりにくい場合は，［画面消去］を押して，白紙の状態から文字のデザインをします．

日本語コード(「リファレンスマニュアル」資料)の中にアルファベットのRがありますから，これを呼び出して，®になおすことにします．

［f・2］(**呼出**)を押します．



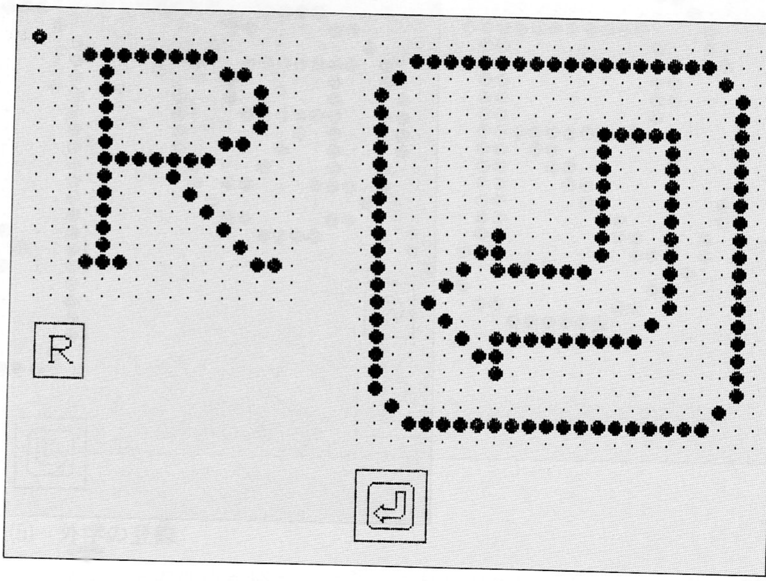
に対して，

2352 ［↵］

と入力します．

注意　RのJISコードです．日本語コード表(「リファレンスマニュアル」資料21ページ参照)で探します．

本機に内蔵されている漢字ROMからRのフォントが読み出されて画面に表示されます．



注意　漢字ROMには，16×16ドットの画面表示用の文字しか入っていません．

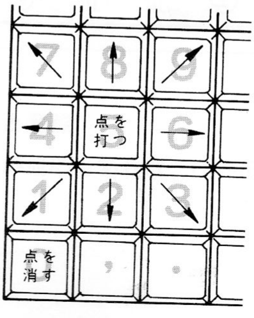
(3)　**16×16ドットのフォントを作る**

16×16のフォント表示エリアの左上端に点滅しているカーソルを動かしながら，Rを®になおしていきます．

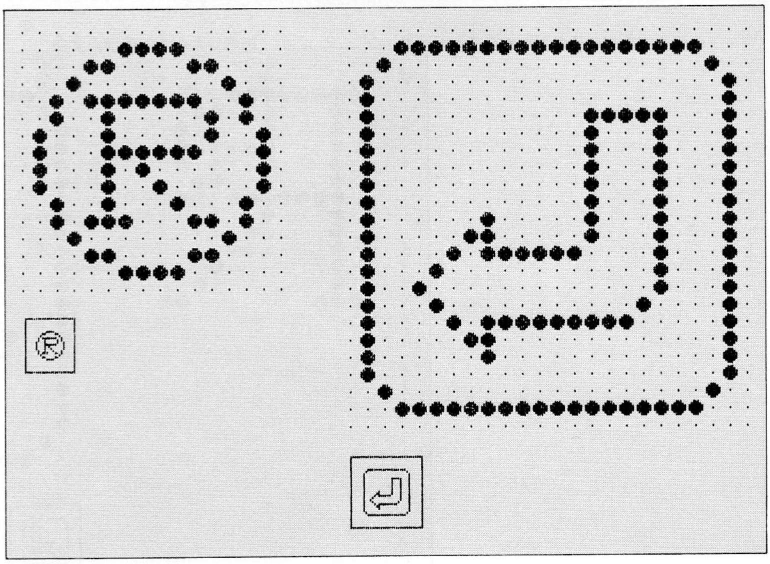
注意　作成中いつでも f･5 （終了）を押すことによって，作成中の外字の定義を中断することができます．
f･5 （終了）を押すと(8)に移ります．

テンキーを使って，カーソルの移動と，ドットのON/OFFを行います．

注意　フォントの作成の途中で初めからやりなおしたいときや，白紙に戻したいときは，作成中いつでも(2)の操作を行うことができます．



下の図は，16×16ドットのフォントができ上がったところです．



(4)　24×24ドットのフォントを作る

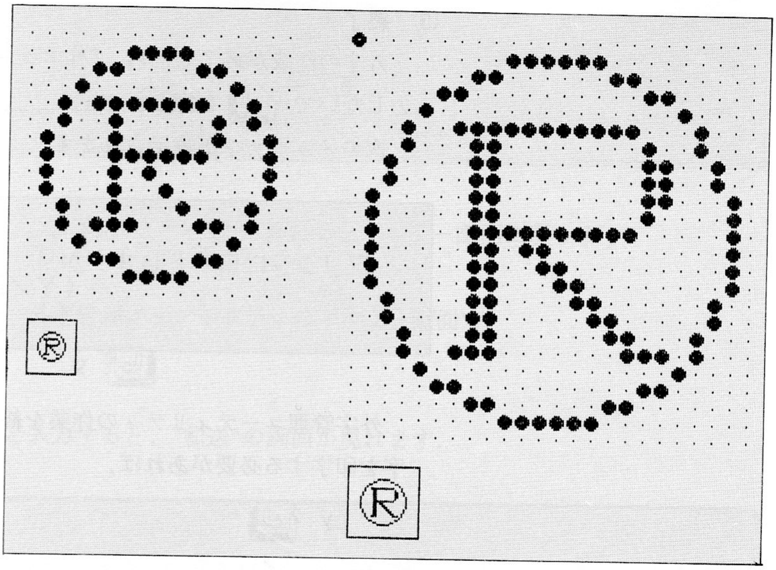
を押すと，カーソルが24×24ドットのフォント表示エリアに移ります．

注意　は，16×16ドットと，24×24ドットの切り替えをします．

24×24ドットのプリンタ印字用の文字も同じようにして作成します．

注意　24×24ドットのフォントは，まずを押して白紙の状態から作りましょう．

次頁の図は，24×24ドットのフォントができ上がったところです．



(5)　外字の登録

フォントの作成が終わったら，これをシステムディスクに書き込みます．

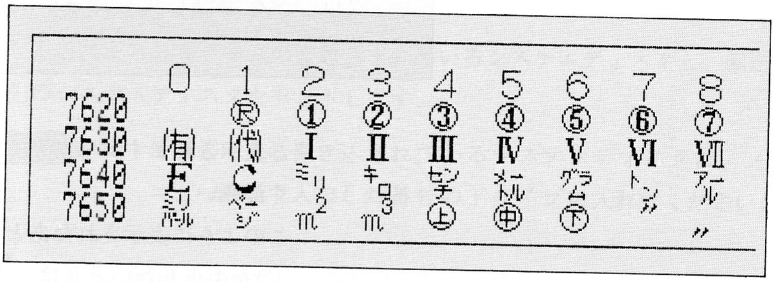
［f・3］（登録）を押すと，書き込みが行われます．

(6)　登録の確認をする

画面下の外字の一覧表を見てみましょう．

注意　標準ディスプレイを使っているときは，［f・4］（一覧）を押すと，外字の一覧表が表示されます．

外字コード7621の欄に，いま作った文字があります．



(7)　他の外字の作成と登録

定義したい外字が他にもあれば，(1)〜(6)で行ったのと同じように繰り返します．

⑻ 終了

外字の作成が終了したら，"外字コード："に対して外字コードを入力しないで ↵ を押します．

次のメッセージが表示されます．

```
プリンタにロードしますか？
                    (Y/N)
```

外字管理ユーティリティの作業を終えたあと，すぐにプリンタに外字を印字する必要があれば，

y ↵

と入力します．

注意 とりあえず外字の作成だけしておき，プリンタへの印字を行わないときはnと入力します．外字をプリンタへ印字する前に，本章の「3　プリンタへの外字ロード」のユーティリティを行ってください．
また，y，nは半角文字で入力してください．大文字でも小文字でもかまいません．以下，同様です．

次のように表示されてプリンタへ外字が書き込まれます．

注意 プリンタが接続されていないと，

```
プリンタを準備してください
```

と表示されますから，プリンタを接続して ↵ を押してください．
このように表示されたあと，プリンタへの書き込み(ロード)を中止したいときは f･5 (**終了**)を押してください．メニュー画面に戻ります．

```
プリンタにロード中です
```

書き込み（ロード）が終わるとメニュー画面に戻ります．

3. 外字を転送する

外字管理ユーティリティのメニュー画面で，

2 ↵

と入力すると，"転送"の画面が現れます．

```
                *****　転　送　*****





転送先のドライブ番号：█

 load                                           終 了
```

もとになる外字データの書き込まれているシステムディスクと，転送先のシステムディスクをセットします．

注意 外字データの書き込まれているシステムディスクは，メニュー画面で入力した番号のドライブに入れてください．

そして，転送先のドライブ番号を，たとえば，

2 ↵

と入力します．

注意 外字の転送にはミニフロッピーディスクドライブが２台以上必要です．

次のように確認を求めてきます．

```
ドライブ 1 ーーー＞ドライブ 2
転送しますか（Ｙ／Ｎ）？█
```

y ⏎

と入力します．

注意 n ⏎ と入力するとメニュー画面に戻ります．

次のように表示しながら，外字データが転送されます．

```
データを転送中です
しばらく　お待ちください
```

転送が終わったら ⏎ を押せば，メニュー画面に戻ります．

3 プリンタへの外字ロード

[prload.fh(PC-8801FH),prload.mh(PC-8801MH)]

このユーティリティは，N88-日本語BASICシステムディスク上にある外字データをプリンタに送ります(ロード)．ただし，外字をロードし印字できるのは，次の日本語プリンタ(またはその同等品)だけです．

※プリンタの電源をOFFにすると，プリンタにロードした外字が消えてしまいます．この場合は，もう1度"prload～"を用いて外字フォントをプリンタにロードしないと，外字は印字できません．

PC-PR201F，PC-PR101F
PC-PR201H2
PC-PR201V
PC-PR201TL
PC-PR101TL
PC-PR406
PC-PR406M

1．プリンタへの外字ロードユーティリティのスタート

N88-日本語BASICをスタートさせ，"prload．～"を実行します．

注意　この例は，8インチディスクユニットを接続していない場合を想定しています．8インチディスクユニットを接続している場合は，ミニディスクドライブのドライブ番号を指定します．

```
run "prload.～" ⏎
```

画面に次のように表示されて，プリンタへのロードが行われます．

注意　必ずミニフロッピーディスクの右側のドライブから外字データが読み出されます．
また，プリンタが接続されていないと，

```
プリンタを準備してください
```

と表示されますから，プリンタを接続して⏎を押してください．

```
外字データをプリンタへロード中です
しばらく　お待ちください
```

2. 終了

プリンタへのロードが終わると，次のようなメッセージを表示して"prload．～"プログラムが終了します．

```
終わりました
Ok
█
```

4 ユーザ辞書管理

[usrdic.fh(PC-8801FH),usrsic.mh(PC-8801MH)]

N_{88}-日本語BASICは，PC-8801MHで約4万語，PC-8801FHで約3万5千語の辞書を持ち，文節単位(PC-8801MH)，熟語単位(PC-8801FH)でかなを漢字に変換しながら，日本語の入力を行うことができます．

さらにN_{88}-日本語BASICは，珍しい名前や長い固有名詞，よく使う決まり文句などを，ユーザ自身の辞書に登録できますので，扱いがとても便利です．

ユーザ辞書管理ユーティリティは，辞書への熟語の登録，変更，削除などを行います．また，ユーザ辞書を，あるシステムディスクから他のシステムディスクに転送することもできます．

1. ユーザ辞書管理ユーティリティのスタート

N_{88}-日本語BASICをスタートさせ，"usrdic.～"をロードし，実行します．

注意 ユーティリティプログラムの入っているフロッピーディスクをドライブ1(8インチディスクユニットが接続されているときはドライブ3)にセットしてください．この例は，ドライブ1からロードしてくると想定したときのものです．

load "usrdic.～" ⏎
run ⏎

しばらくすると次のように表示されます．

```
変更したいユーザ辞書を含むシステムディスクを
ドライブ　1　にセットしてください
```

対象となるユーザ辞書の入っているシステムディスクをドライブ1にセットし，⏎を押します．

注意 システムディスクには，ユーザ辞書用の領域があらかじめ確保されています(買ったばかりのときには空っぽですが)．ユーザ辞書はFILES命令では見ることはできません．

メニュー画面が現れます．

```
《ユーザ辞書》

1．登 録 ....新規に熟語を登録
2．変 更 ....登録した熟語の変更
3．削 除 ....登録した熟語の削除
4．一 覧 ....ユーザ辞書の画面表示および，プリンタ出力
5．転 送 ....ユーザ辞書を別のメディアに転送

どの処理を行いますか？

load                                              終 了
```

2．登録

(1)　メニュー画面で 1 ↵ と入力します．

(2)　次のように，読みがなの入力を求めてきます．

```
読み：█
```

ここで，辞書に登録したいことばの変換のもとになる「読み」をひらがなで入力し，↵ を押します．

注意　「読み」は，全角文字のひらがな，英数字で入力します．入力の方法がわからないときは，第5章をご覧ください．
また，熟語の登録を中断してメニューに戻りたいときは，ここで何も入力しないで ↵ を押してください．

ここでは，例として，

にちでん ↵

と入力してみます．

(3)　次に，「読み」に相当する語を入力します．

```
熟 語：█
```

に対して，ここでは，

日本電気株式会社 ↵

と入力しましょう．

注意　熟語は全角文字16文字以下で入力してください．
そして，熟語の登録を中断してメニュー画面に戻りたいときは，ここで何も入力しないで ↵ を押してください．
また，すでに同じ語が登録してあれば，

```
同じ熟語が既に登録されています
```

というメッセージが表示されます．

(4)　「読み」と「熟語」の入力が終わると，辞書に入れてよいかどうかを確認してきます．

```
登録しますか（Y／N）？█
```

y ↵

と入力します．

注意　**n** ↵ と入力すると，(2)に戻ります．

(5)　(2)に戻りますから，他に登録したい読みがあれば(2)〜(4)を繰り返し，登録を終了するときは"読み："に対して ↵ を入力すれば，メニュー画面に戻ります．

3. 変更

(1) メニュー画面で2 と入力します.

```
読み：█
```

変更したい語の「読み」を入力します.

注意 読みを入力せずに を押すと，変更は中断されてメニュー画面に戻ります.

(2) 入力した「読み」で登録してある「熟語」がすべて表示されます.

注意 登録してある熟語が一画面に表示しきれないときは， を押すと続きが表示されます.

```
              ＊＊＊＊＊変　更＊＊＊＊＊
読み：にちでん

          1. 日本電気株式会社
          2. 日本電気
          3. 日電
      登録語数＝ 3

どの熟語を変更しますか？█

 load                                    終了
```

変更したい「熟語」の番号を入力します.

注意 「読み」に対する「熟語」が一語だけのときは，番号を入力する必要はありませんから(3)に移ります.

たとえば，上の例の2番を変更したければ，

2

と入力します.

(3) 新しい熟語を入力します．

注意 熟語を入力せずに ⏎ を押すと，変更は中断されてメニュー画面に戻ります．

```
熟 語：█
```

(4) 新しい熟語を入力し終わると，

```
"日本電気"ーーー＞"日電ＨＥ"に変更しますか（Ｙ／Ｎ）？█
```

と確認を求めてきますから，

y ⏎

と入力します．

注意 n ⏎ と入力すると，熟語の変更は中断されて(1)に戻ります．

(5) 辞書が書き換えられて(1)に戻ります．

注意 変更を終了したいときは，"読み："と表示されている状態で，読みを入力しないで ⏎ を押せばメニュー画面に戻ります．

4. 削除

(1) メニュー画面で **3** と入力します.

```
読み:█
```

削除したい「熟語」の読みを入力します.

注意 読みを入力せずに を押すと，削除は中断されてメニュー画面に戻ります.

(2) 入力した"読み"で登録してある"熟語"がすべて表示されます.

注意 登録してある熟語が一画面に表示しきれないときは，を押すと続きが表示されます.

```
             *****削　除*****
読み:にちでん
          1. 日本電気株式会社
          2. 日電
          3. 日電HE
      登録語数＝　3

どの熟語を削除しますか？█

 load                                      終　了
```

削除したい"熟語"の番号を入力します.

注意 "読み"に対する"熟語"が一語だけのときは，番号を入力する必要はありませんから(3)に移ります.

たとえば，上の例の2番を削除したければ，

2

と入力します.

(3)　次のように確認を求めてきます.

```
                    ＊＊＊＊＊ 削　除 ＊＊＊＊＊
読 み：にちでん
                  1. 日本電気株式会社
                  2. 日電
                  3. 日電ＨＥ
            登録語数＝　3



日電 を削除しますか（Ｙ／Ｎ）？
 load                                              終　了
```

y

と入力します.

注意　n　と入力すると，熟語の削除は中断されて(1)に戻ります.

(4)　辞書が更新されて(1)に戻ります.

注意　"読み"に相当する"熟語"が1つしかなかった場合は，"読み"も"熟語"と同時に削除されます.
また，削除を終了したいときは，"読み："と表示されている状態で，読みを入力しないで　を押せばメニュー画面に戻ります.

5. 一覧

(1) メニュー画面で**4** ↵と入力します.

(2) プリンタへも同時に出力するかどうかを選びます.

注意 ↵だけを入力するとメニュー画面に戻ります.

```
プリンタに出力しますか（Y／N）？█
```

画面と，プリンタとの両方に出力したければ，

y ↵

と入力します.

(3) ユーザ辞書に登録されている語をすべて見たいか，"読み"を指定したいかを**1** ↵または**2** ↵で選択します.

注意 ↵だけを入力するとメニュー画面に戻ります.

```
＊＊＊＊＊ 一　覧 ＊＊＊＊＊

1．全　部 ....ユーザ辞書を全部表示
2．部　分 ....指定した読みの熟語のみ表示

どの処理を行いますか？█
```

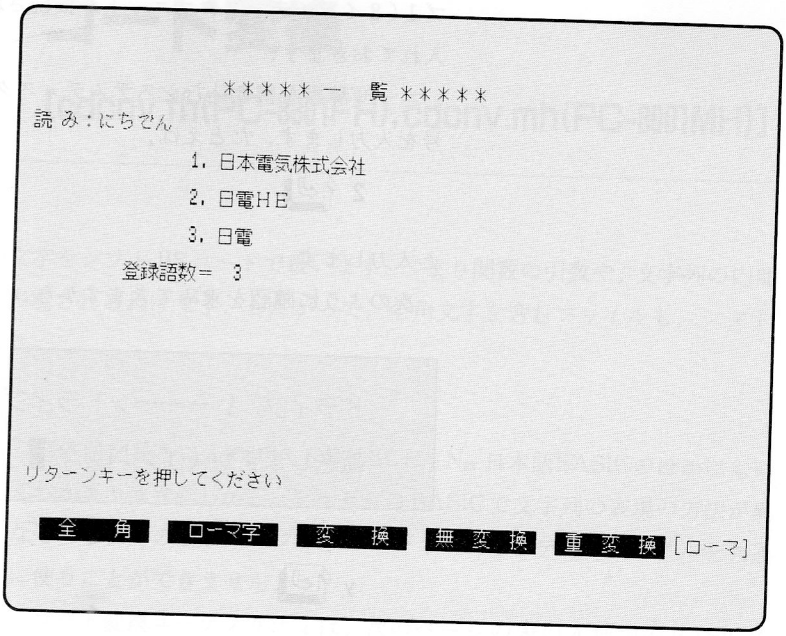
(4) (3)で**2** ↵を押した場合は，

```
読 み：█
```

に対して「読み」を入力します.

注意 ↵だけを入力するとメニュー画面に戻ります.

(5) 一覧表が表示されます．



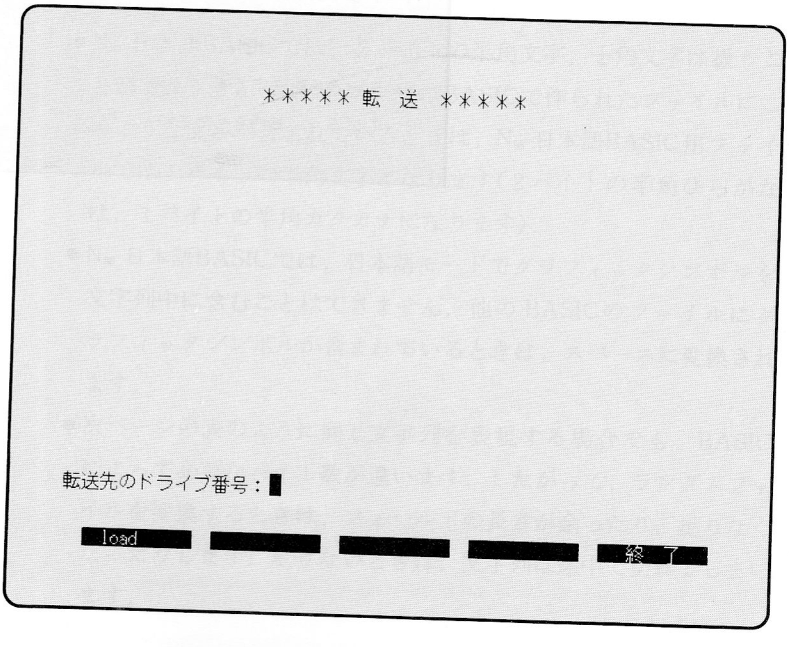
一画面に入りきらないときは，[↵]を押しながら表示させていきます．

一覧表の表示が終わったら，[↵]を押すとメニュー画面に戻ります．

注意 (3)で"1.全部"を選択した場合，途中で表示を中断したいときは [f･5]（**終了**）を押します．

6. 転送

メニュー画面で**5** [↵] と入力します．



もとになるユーザ辞書の入っているシステムディスクは，必ずドライブ1(8インチディスクユニットが接続されているときはドライブ3)に入れておきます．

ユーザ辞書を書き込むシステムディスクをセットし，そのドライブ番号を入力します．たとえば，

2 ⏎

と入力します．

次のように確認を求めてきますから，

```
ドライブ 1 ―――>ドライブ 2
転送しますか（Y／N）？■
```

y ⏎

と入力してください．

注意 n ⏎ と入力するとメニュー画面に戻ります．

次のように表示されて転送が行われ，終了するとメニュー画面に戻ります．

```
＊＊＊＊＊ 転 送 ＊＊＊＊＊
ドライブ 1 ―――>ドライブ 2

データを転送中です
しばらく お待ちください
```

5 コード変換
[cdcnv.fh(PC-8801FH),cdcnv.mh(PC-8801MH)]

N88-日本語BASICでは，全角文字をシフトJISコードで扱います．つまり関数の引数や，文字列の内部形式などがシフトJISコードで表現されるわけです．したがって，全角文字を含むファイルも，シフトJISコードから成り立っています．

日本語を扱うことのできるBASICは，N88-日本語BASICのほかにもいくつかあります．しかし，それぞれのBASICで文字列の表現の方法が異なるため，そのままではフロッピーディスク上のデータファイルを共通に使うことができません．

コード変換ユーティリティは，次のページの表の4種類のBASICで作成したデータファイルに含まれる，異なった表現の全角文字を互いに変換します．

コード変換ユーティリティは，データファイル(シーケンシャルファイルとランダムファイル)，およびアスキーセーブされたプログラムファイルに含まれる文字列を変換します．バイナリセーブされたプログラムファイルや機械語ファイルの変換は行いません．

各BASICで文字列の表現が異なるために，次のような制限があります．

- N88-日本語BASICでは，2バイトの半角文字，$\frac{1}{4}$角文字は扱うことができません．このため，他のBASICで作られたファイルにこのような文字が含まれているときは，N88-日本語BASIC用ファイルでは1バイトの半角文字になります(2バイトの半角ひらがなは，1バイトの半角カタカナになります)．
- N88-日本語BASICでは，日本語モードでグラフィックシンボルを文字列中に含むことはできません．他のBASICのファイルにグラフィックシンボルが含まれているときは，スペースに変換されます．
- 次ページの表のように同じ文字列を表現する場合でも，BASICによって必要なバイト数が違います．したがって，ランダムファイルを変換するときは，フィールドの長さが余ったり，足りなくなったりします．足りないときは，文字列は途中で切れてしまいます．

BASIC名(発売元)	対象機種	全角文字の表現 (例として"AB日本語CD"を16進数で表現した)	コード変換ユーティリティで扱えるフロッピーディスクのタイプ
N$_{88}$-日本語BASIC (日本電気株式会社)	PC-8801MH PC-8801FH PC-8801mkⅡMR/ PC-8801mkⅡFR/ PC-8801mkⅡSR用	シフトJISコードを使う．全角文字は2バイト，半角文字は1バイトで表現される． (例) A｜B｜日｜本｜語｜C｜D 41｜42｜93FA｜967B｜8CEA｜43｜44 10バイト	●5.25インチ両面高密度，両面倍密度ミニフロッピーディスク(2HD, 2D) ●8インチ両面倍密度フロッピーディスク
N$_{88}$-日本語BASIC(86) (日本電気株式会社)	PC-9800シリーズ用	JISコードの前にKI(漢字イン)，後にKO(漢字アウト)を付ける． (例) A｜B｜KI｜日｜本｜語｜KO｜C｜D 41｜42｜1B4B｜467C｜4B5C｜386C｜1B48｜43｜44 14バイト	●5.25インチ両面高密度ミニフロッピーディスク(2HD) ●5.25インチ両面倍密度倍トラックミニフロッピーディスク(2DD) ●5.25インチ両面倍密度ミニフロッピーディスク(2D) ●8インチ両面倍密度フロッピーディスク
8801漢字BASIC (株式会社システムソフト) (8801漢字BASICは株式会社東海クリエイトとの協同企画商品です.)	PC-8800シリーズ用	JISコードの，8ビット目と16ビット目をONにしたコードの前後に漢字イン，漢字アウトを付ける．2バイトの半角文字，1/4角文字も扱うことができる． (例) A｜B｜KI｜日｜本｜語｜KO｜C｜D 41｜42｜7F｜C6FC｜CBDC｜B8EC｜7F｜43｜44 12バイト	●5.25インチ両面倍密度ミニフロッピーディスク(2D) ●8インチ両面倍密度フロッピーディスク
N$_{88}$-漢字BASIC (PC-8834-2W(K)) (日本電気株式会社)	PC-8800シリーズ用	JISコードを，下位バイト，上位バイトの順に並べ，前にKIを付ける．2バイトの半角ひらがなと，1バイトの半角文字の前にはそれぞれ特殊な制御文字（下の例の＊の文字）を付ける． (例) ＊｜A｜B｜KI｜日｜本｜語｜＊｜C｜D FD｜41｜42｜FA｜7C46｜5C4B｜6C38｜FD｜43｜44 13バイト	●5.25インチ両面倍密度ミニフロッピーディスク(2D) ●8インチ両面倍密度フロッピーディスク

1. コード変換ユーティリティのスタート

N$_{88}$-日本語BASICをスタートさせ，"cdcnv.～"をロードし，実行します．

```
load "cdcnv.～"⏎
run⏎
```

しばらくすると，メニュー画面が表示されます．

```
《 N88-日本語BASICコードコンバータ 》

N88-日本語BASICと他の漢字BASICの間の日本語コード変換を行います

1. 8801漢字BASIC--->N88-日本語BASIC
2. N88-漢字BASIC ---->N88-日本語BASIC
3. N88-日本語BASIC(86)-->N88-日本語BASIC
4. N88-日本語BASIC --->8801漢字BASIC
5. N88-日本語BASIC --->N88-日本語BASIC(86)

どの変換を行いますか？■

load                                    終 了
```

2. メニューの選択

1.～5.のうちから，どの変換を行うかを選択して番号を入力します．ここでは「N_{88}-漢字BASIC→N_{88}-日本語BASIC」を例に，操作手順を説明していきます．

2 ⏎

と入力すると，次のように表示されます．

注意 ⏎ だけを入力すると終了します．

```
N88漢字BASIC-->N88日本語BASIC

N88漢字BASICのドライブ番号は？

load                                    中 止
```

3. データの変換元のドライブ番号の指定

ここで，変換されるデータファイルの入っているドライブ番号を入力します．ここではドライブ1にN_{88}-漢字BASICのファイルが入っていると想定します．

1 ⏎

と入力します．

注意 ********************
PC-8801MHの場合のみ……
****
1.で3.を選んだときは，この後，フロッピーディスクのタイプを指定します．

```
フロッピーディスクのタイプは（1:2D, 2:2DD, 3:2HD）？█
```

に対して1～3の番号を入力してください．

4. 変換したデータの転送先のドライブ番号の指定

次に，変換されたファイルを転送する先のドライブ番号を答えます．

```
N88-日本語ＢＡＳＩＣのドライブ番号は？█
```

ここでは，

2 ⏎

と入力します．

注意 ********************
PC-8801MHの場合のみ……
****
転送する先のドライブ番号は，もとになるファイルの入っているドライブ番号とは違う番号を指定しなければなりません．
そして，1.で5.を選んだときは，この後，フロッピーディスクのタイプを指定します．

```
フロッピーディスクのタイプは (1:2D, 2:2DD, 3:2HD) ?█
```

に対して1～3の番号を入力してください.

****

PC-8801FHの場合のみ……

****

転送のもとになるファイルが入っているのと同じドライブ番号を指定すると，5.のときに(この例ではドライブ1)変換されてできるファイルのファイル名も聞いてきます.

つまり，同じフロッピーディスクに，文字表現の異なる2つのデータファイルを持つことになります.

5. 変換するデータファイルの選択

ドライブ1に書き込まれているすべてのファイル名が表示され，この中からどのデータファイルを変換したいかを聞いてきます.

```
N88-漢字BASIC——>N88-日本語BASIC

鬼島   .J      海老   .        kpload.       J TO S.J      j to s.j
SJ TO  .J      ADINST.R       ENINST.R      GBCHAP.2      data1  dat
abc1

変換するファイル名：

[ load ]  [      ]  [      ]  [      ]  [ 中  止 ]
```

ファイル名を，たとえば

data1.dat ⏎

のように入力します.

注意　BASICによっては，ファイル名に全角文字を使用できない場合がありますから注意してください．
また，ドライブ1にN_{88}-日本語BASICシステムディスクがセットされていないと，全角文字の入力はできません．
コード変換の対象となるファイルは，半角文字でファイル名を付けることをお勧めします．

6. ランダムファイルの変換

次に，

```
ランダムファイルですか（Y／N）？ █
```

と聞いてきます．ランダムファイルを変換する場合は，

y ⏎

と入力します．

注意　シーケンシャルファイル(プログラムをアスキーセーブしたファイルも含む)の場合は **n** ⏎ と入力します．

7. フィールドの指定

ランダムファイルの場合は，フィールドがどのように指定されているデータであるかを答える必要があります．

```
フィールドの指定：
```

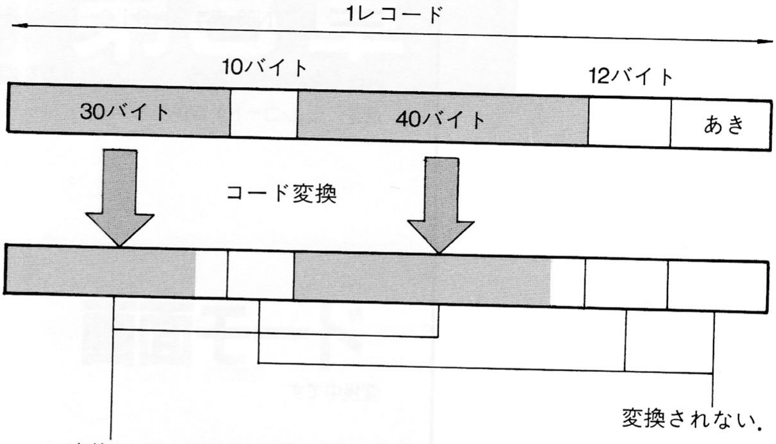
に対して，コード変換をしたい文字の部分のフィールドの長さと，数値の部分とに区別して，たとえば次のように指定します．

C30,　N10,　C40,　N12 ⏎

文字列がそれぞれ30バイト，40バイトの長さでフィールド設定されていることを示す．

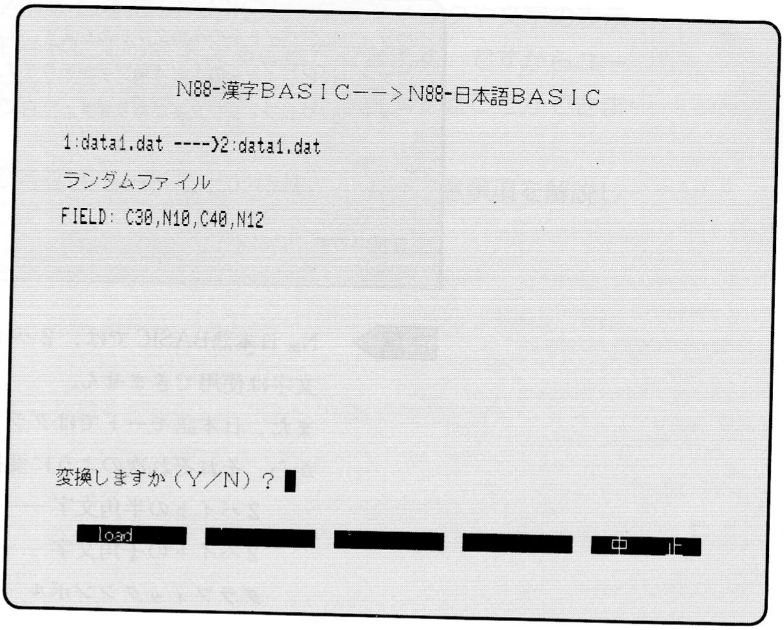
数値がそれぞれ10バイト，12バイトの長さでフィールド設定されていることを示す．

注意　図示すると，次のようになります．



8. 確認

変換を始めてよいかどうか確認を求めてきます．



y ⏎

と入力すると，変換が始まります．

注意　n ⏎ と入力すると，メニュー画面に戻ります．

変換中は，次の例のようにメッセージが表示されます．

```
N88-漢字BASIC——>N88-日本語BASIC

変換するファイルは全部で 5 レコードあります
現在  2 レコードが変換済みです



変換中です

load                                   中  止
```

ファイルに，グラフィックシンボルや，2バイトで表されている半角文字（$\frac{1}{4}$角文字）があった場合，次のような警告を表示します．

```
ファイルに2バイト系の半角文字があります。1バイト系の文字に変換しました
ファイルに2バイト系の1／4角文字があります。1バイト系の文字に変換しました
ファイルにグラフィック文字があります。空白に変換しました


変換中です
```

注意 N88-日本語BASICでは，2バイトで表される半角文字や$\frac{1}{4}$角文字は使用できません．
また，日本語モードではグラフィックシンボルは扱えませんから，それぞれ次のように変換されます．

2バイトの半角文字 ——→ 1バイトの半角文字
2バイトの$\frac{1}{4}$角文字 ——→ 1バイトの半角文字
グラフィックシンボル ——→ スペース

変換が終わると次のように表示されます．⏎を押すとメニュー画面に戻ります．

```
変換は終了しました
```

第8章

画面モード

N88-BASIC/N88-日本語BASICは，グラフィックや文字の表示能力を，画面モードによって変えることができます．ですからユーザはその時々に応じて，最適な画面モードを選択することができます．
この章ではこの各画面モードごとの特性，および注意事項を解説していきます．

1 画面モードの種類と各画面モードの特色

1．画面モードの意味

ここでいう画面モードとは，ドットグラフィックスの分解能，白黒モードかカラーモード上で文字を表示するテキスト画面の表示桁数など画面に対する操作のすべてを含むものです．

N_{88}-BASIC/N_{88}-日本語BASICは，この画面モードを必要に応じて選ぶことができます．

2．画面モードの種類

N_{88}-BASIC/N_{88}-日本語BASICは，次の5つの画面モードを持っています．

〈画面モード一覧〉

画面モード	パラメータ	解像度	使用できる画面	設定方法
グラフィックシンボルモード	0	ヨコ タテ 640 × 200	・テキスト ×1 ・カラーグラフィック×1	SCREEN 0
	1	640 × 200	・テキスト ×1 ・白黒グラフィック ×3	SCREEN 1
	2	640 × 400	・テキスト ×1 ・白黒グラフィック ×1	SCREEN 2
日本語モード	3	640 × 200	・テキストとカラーグラフィック兼用画面 ×1	SCREEN 3
	4	640 × 400	・テキストとグラフィック兼用の白黒画面 ×1.	SCREEN 4

（ただし画面モード2，4は，専用高解像度ディスプレイが接続されているときのみ使用できます．）

また観点を変えて，グラフィック表示がカラーで行えるか否か，並びに使用しているBASICの種類によって分類すると，次のようになります．

解像度＼グラフイック表示	白黒モード	カラーモード
ヨコ タテ 640 × 200	SCREEN 1 (N_{88}-BASIC)	SCREEN 0 (N_{88}-BASIC) SCREEN 3 (N_{88}-日本語BASIC)
ヨコ タテ 640 × 400	SCREEN 2 (N_{88}-BASIC) SCREEN 4 (N_{88}-日本語BASIC)	

3. 画面モードの設定方法

5つの画面モードのうち，どれか特定のものを選択したい場合は，前ページ画面モード一覧の右端に示したように，SCREEN文で設定します．

なお，初期設定はN_{88}-BASICでは画面モード0（SCREEN 0），N_{88}-日本語BASICにおいては，接続されているディスプレイの解像度により，画面モード3（SCREEN 3），あるいは画面モード4（SCREEN 4）のいずれかを選びます．これを図示すると次のとおりです．

	標準ディスプレイ	高解像度ディスプレイ
N_{88}-BASIC	モード0（SCREEN 0）	モード0（SCREEN 0）
N_{88}-日本語BASIC	モード3（SCREEN 3）	モード4（SCREEN 4）

4. 各画面モードの特色

画面モード0……このモードは，640×200ドットの解像度を持ち，カラーグラフィック表示が行えます．また，テキスト画面とグラフィック画面が分かれているので，それぞれを独立して操作することができます．

画面モード1……このモードは，テキスト画面とは別にグラフィック画面を白黒3画面分持ち，任意の複数のグラフィック画面を合成，表示することができます．また，テキスト画面とグラフィック画面が分かれているので，それぞれ独立に操作することができます．

画面モード2……このモードは，640×400ドットの高解像度白黒グラフィック表示が行えます．また，テキスト画面とグラフィック画面が分かれているので，それぞれ独立に操作することができます．

画面モード3……このモードは640×200ドットの解像度を持ち，カラーグラフィック表示が行えます．モード0との違いは，日本語表示が行える代わりに，テキスト画面とグラフィック画面とが分離，独立していない点です．

画面モード4……このモードでは，640×400ドットの高解像度白黒グラフィック表示が行えます．モード2との違いは，日本語表示が行える代わりに，テキスト画面とグラフィック画面とが分離，独立していない点です．

5. BASICで使用可能な画面モード

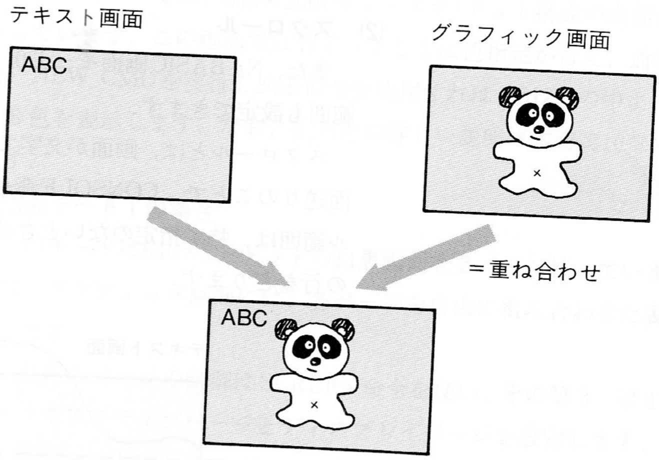
N_{88}-BASICでは，画面モード0，1，2の3種が使えます（画面モード3，4は使えません）．

N_{88}-日本語BASICでは，5種すべての画面モードが使えます．

2 画面モード0，1，2の表示

1．画面構成

これらのモードは，テキストとグラフィックという２つの画面を持っています．テキスト画面というのは文字を表示するときに使うもので，スクリーンエディタより入力した文字やプログラムを表示するときに使います．一方，グラフィック画面はCIRCLEやLINE等のグラフィックコマンドを実行したときに使う画面です．なお，この２つの画面はN_{88}-BASIC/N_{88}-日本語BASICの内部的なもので，ディスプレイには両画面を重ねたものが表示されます．



N_{88}-BASIC/N_{88}-日本語BASICでは，この２つの画面は完全に独立しています．そのうえ画面ごとにモードを切り替える命令があるので，画面どうしの影響はありません．

以下に，テキスト画面，グラフィック画面の順に，それぞれの特性をご紹介します．

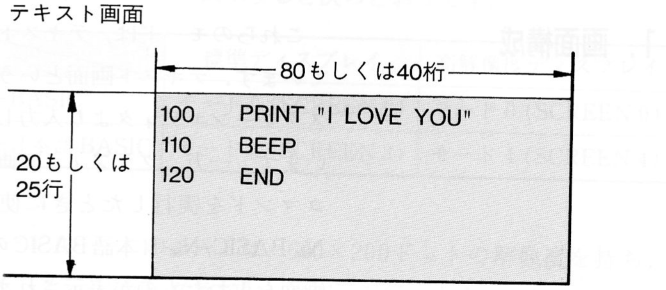
2．テキスト画面

(1) 表示桁数の指定

テキスト画面の表示文字数は横80桁か40桁で，縦20行か25行を選択できます．

これらの設定はWIDTH文を使って行います．なおN_{88}-BASICの立ち上げ時には80桁×20行に設定されます．

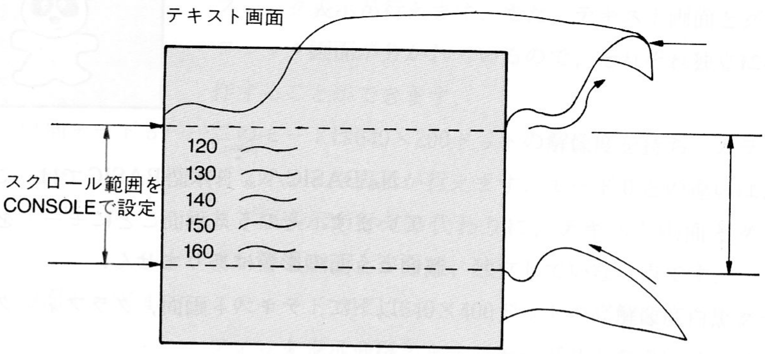
認定方法	表示文字数(横×縦)
WIDTH 40, 20	40×20
WIDTH 80, 20	80×20
WIDTH 40, 25	40×25
WIDTH 80, 25	80×25



(2) スクロール

また，N₈₈-BASIC画面モード0，1，2ではテキスト画面のスクロール範囲も設定できます．

スクロールとは，画面が文字でいっぱいになったときに行われる画面送りのことで，CONSOLE命令で設定します．なお，このスクロール範囲は，特に指定のないときは初期値としてテキスト画面のすべての行をとります．



(3) 白黒／カラーモードの選択

文字の表示の仕方はCONSOLE文を使い，白黒モード／カラーモードの選択ができます．

このときの表示指定(アトリビュート指定)はCOLOR文を使い，1文字ごとに表示色や点滅表示の有無を指定します．また，すでに表示した文字については，COLOR@命令を使うことにより領域ごとに表示の変更ができます．

指定方法	白黒モード CONSOLE, , , 0	カラーモード CONSOLE, , , 1
COLOR文 第1パラメータ および COLOR@文 第3パラメータ の意味	0－ノーマル(通常の表示) 1－シークレット (文字を表示しない) 2－ブリンク(点滅する) 3－ブリンクシークレット(1と同じ) 4－リバース(反転する) 5－リバースシークレット (反転して文字は表示されない) 6－リバースブリンク (反転して点滅する) 7－リバースブリンクシークレット (5と同じ)	0－黒 1－青 2－赤 3－紫 4－緑 5－水色 6－黄色 7－白

3. グラフィック画面

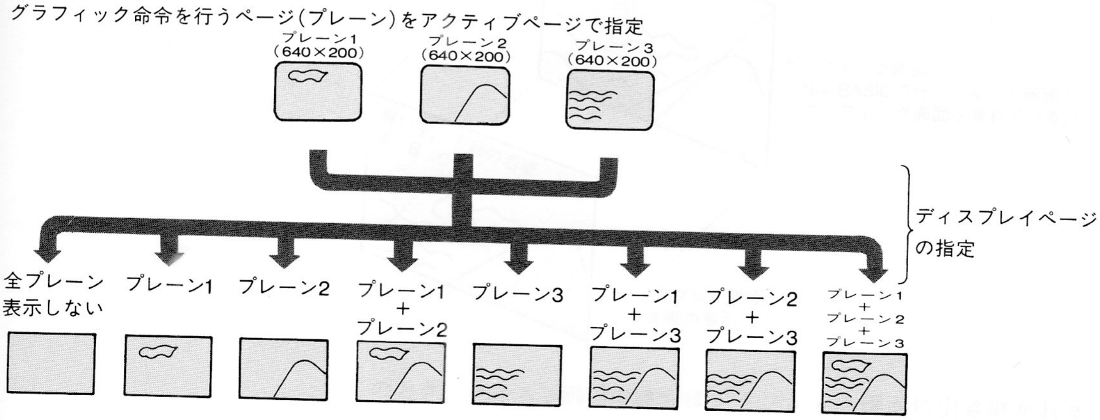
(1) 画面モード0の場合

カラーモードでは，640×200(ヨコ×タテ)ドット構成の画面を1つ持っています．この画面では各ドットごとに色指定が行え，通常は8色，NEW CMDを実行し拡張命令を使用すれば，512色の中より選んだ8色を表示します．なお，表示色の詳しい説明は本書第10章で行います．

(2) 画面モード1の場合

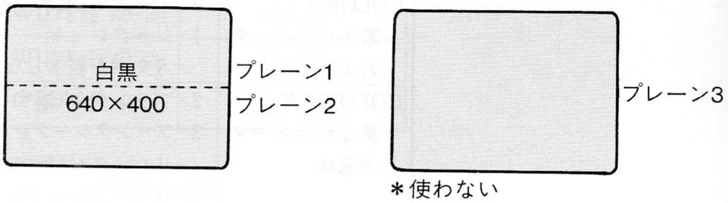
このモードは640×200ドットの白黒画面を3ページ持っています．なお，グラフィック画面には，各ページを自由に組み合わせた表示ができます．

これらのページの制御はSCREEN命令を使い，その第3，第4パラメータでアクティブページとディスプレイページを設定します．アクティブページとは，グラフィック命令により描画を行うページのことで，ディスプレイページとは，グラフィック画面に表示を行うページのことです．



(3) 画面モード２の場合

640×400ドットの白黒画面を１ページ持つモードです．



専用ディスプレイモード

画面の構成は図のように，640×200白黒モードのときのプレーン１とプレーン２が２つ上下につながった形となっています．このモードではプレーン３は使いません．

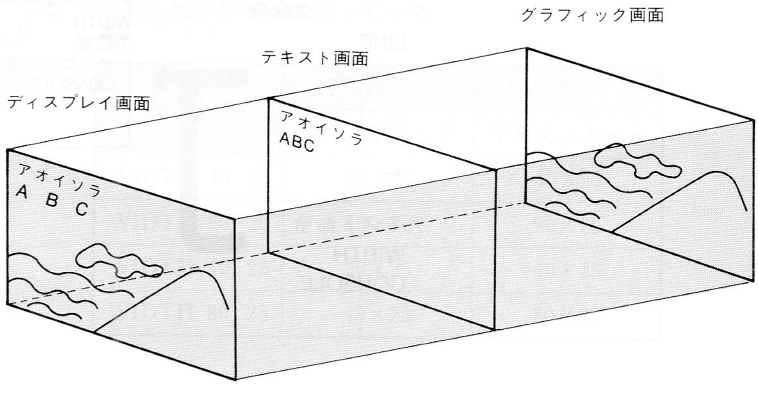
ただし，このモードは専用高解像度ディスプレイを使用したときのみ設定可能となります．

3 画面モード3,4の表示

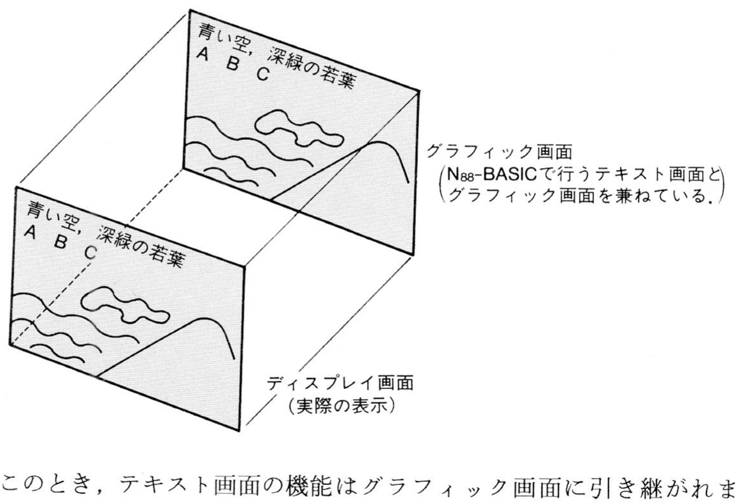
1. 画面構成

画面モード3，4では，日本語処理を行えますが，モード0，1，2と異なりテキスト画面を持っていません．このとき，テキストはグラフィック画面に，点や線のグラフィックと一緒に表示されます．これは，文字とグラフィックは相互に影響しあうことを意味します．ですから，同じ場所に文字とグラフィックを表示したときには，上書きが行われて最初に書かれた方は消えるので，注意が必要です．

モード0, 1, 2の画面構成

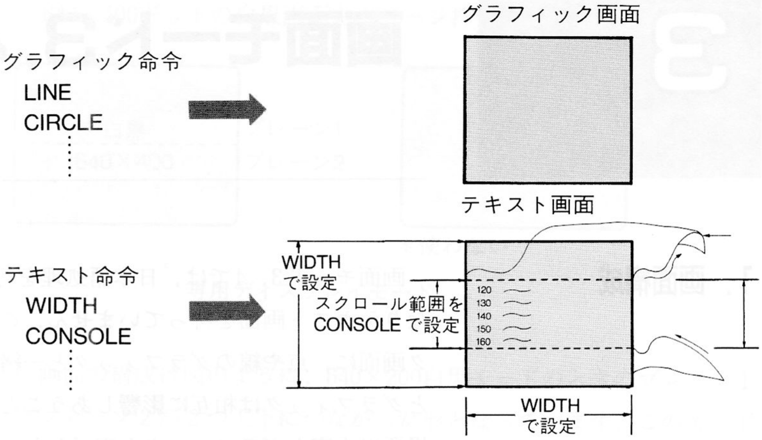


モード3, 4の画面構成

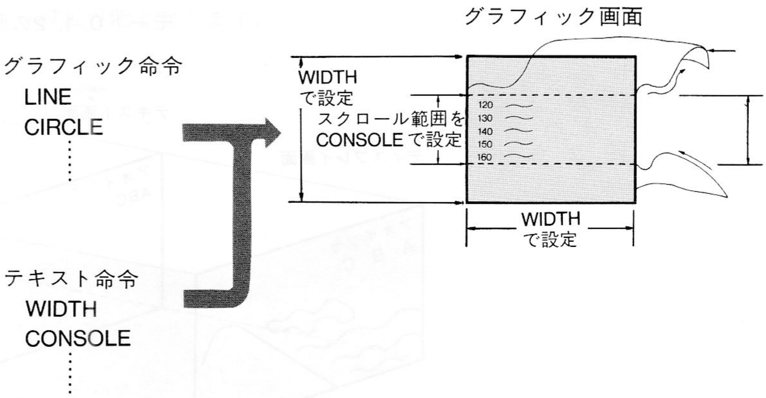


このとき，テキスト画面の機能はグラフィック画面に引き継がれます．これを図に示すと次ページのようになります．

モード0, 1, 2



モード3, 4



2. テキストの表示

(1)　**画面モード3**

このモードでは，次の範囲内で表示文字数を設定するほかはモード0と同じですが，半角文字モードと全角文字モードでは表示文字数が異なります(全角文字モード＝半角文字モード×２).

設定方法	表示文字数	
	全角文字モード	半角文字モード
WIDTH 40, 10	20×10	40×10
WIDTH 40, 12	20×12	40×12
WIDTH 80, 10	40×10	80×10
WIDTH 80, 12	40×12	80×12

(2)　**画面モード4**

このモードは白黒画面ですが，最大40桁×25行(ヨコ×タテ)の範囲内で，日本語表示を行います.

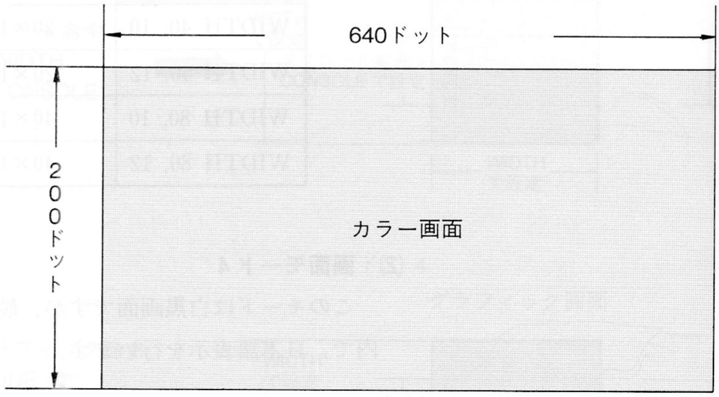
設定方法	表示文字数	
	全角文字モード	半角文字モード
WIDTH 40, 20	20×20	40×20
WIDTH 80, 20	40×20	80×20
WIDTH 40, 25	20×25	40×25
WIDTH 80, 25	40×25	80×25

3. グラフィックの表示

(1) 画面モード3の場合

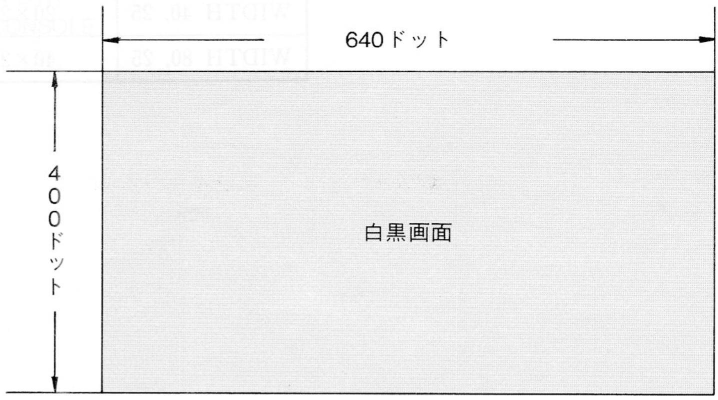
このモードは，画面モード0と同様に640×200(ヨコ×タテ)ドットの構成の画面を1つ持っています．

この画面では各ドットごとに色指定が行え，通常は8色，NEW CMDを実行し拡張命令を使用すれば，512色の中より選んだ8色を表示します．なお，表示色の詳しい説明は本書第10章で行います．



(2) 画面モード4の場合

このモードは画面モード2と同様に，640×400(ヨコ×タテ)ドットの白黒画面1ページより構成されます．



第9章

グラフィックス

N88-BASICは本格的なグラフィック処理を実現するために，仮想描画空間という概念を導入しています．これは，コンピュータシステムにワールド空間という仮想の描画空間を設け，ここにグラフィックを行おうというものです．描かれたグラフィックはN88-BASICによりグラフィック画面へ変換，表示が行われます．これにより，ワールド空間に描いたグラフィックを任意に拡大，縮小，移動表示することが実現しました．

1 座標系の概要

1. 座標系の種類と各座標系の関係

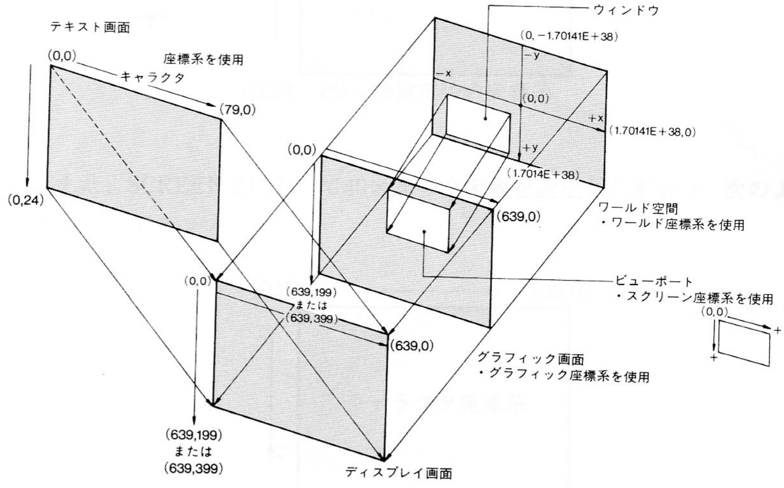
N88-BASIC/N88-日本語BASICは，文字の位置指定のための座標系と，グラフィック描画のための座標系の2つの座標系を備えています．

前者を，キャラクタ座標系といいます．

後者を，グラフィック座標系といいます．

グラフィック座標系は，さらにスクリーン座標系とワールド座標系という2つの座標系を持っています．

各座標系の関係は，概念的に図示すれば次のとおりです．



2. 各座標系の機能

キャラクタ座標系——テキスト画面に存在する座標です．座標の1点は1キャラクタ(キーボードを打って出る文字のこと)に対応します．

グラフィック座標系—グラフィック画面に存在する座標です．座標の1点はディスプレイ上のドットに対応しています．

ワールド座標系———BASICが設定した架空の座標です．ほとんどのグラフィック命令はこの座標系に対して働きます．

スクリーン座標系——ビューポートをグラフィック画面に設定したとき，ビューポート内に存在する座標です．

※ オリジナルスクリーン座標系は，ディスプレ

イ画面のすべてをビューポートと考えたときの

スクリーン座標系.

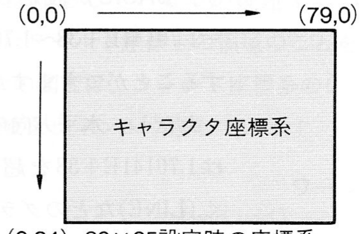
3. 各座標系に有効なBASIC命令

- 座標系
  - キャラクタ座標系
    - ・LOCATE
  - グラフィック座標系
    - ワールド座標系
      - WINDOW
      - PSET
      - PRESET
      - LINE
      - CIRCLE
      - PAINT
    - スクリーン座標系
      - GET@
      - PUT@
      - POINT(関数)
      - ※オリジナルスクリーン座標系に対して働くもの
      - VIEW

2 各座標系のサイズ

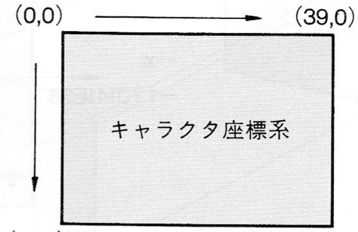
1. キャラクタ座標系

キャラクタ座標系は，テキスト画面に展開される座標系です．WIDTH文による表示桁，行数の設定に従い，キャラクタ座標系の大きさは変動します．たとえば80×25の文字数を設定した場合，座標の位置は次の図のようになります．



80×25設定時の座標系

また，SCREEN 2において40×20の文字数を設定した場合は，次のようになります．

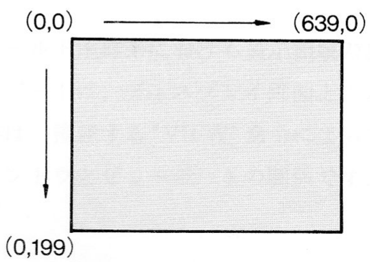


40×20設定時の座標系

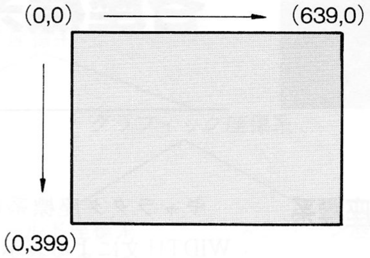
2. グラフィック座標系

グラフィック座標系はグラフィック画面に展開される座標系です．

画面モード0，1，3(解像度640×200)のときは，カラー，白黒を問わず次のようになります．



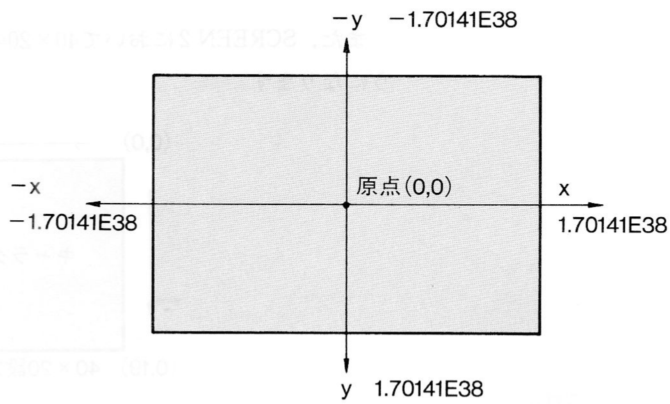
また，画面モード２，４(解像度640×400)のときは，次のようになります．



3. ワールド座標系

ワールド座標系とは，ユーザが使うことのできる最大の座標系であり，BASICが設定した架空の座標です．その大きさは，縦・横とも−1.70141E+38〜1.70141E+38まで許されていて，負の領域座標も表現することができます．

ただし，水平方向の幅(WX_2-WX_1)および垂直方向の幅(WY_2-WY_1)は1.70141E+38を超えてはなりません．点を打つ(PSET)，線を引く(LINE)などのグラフィック命令のほとんどは，このワールド座標系に対して働きます．



4. スクリーン座標系

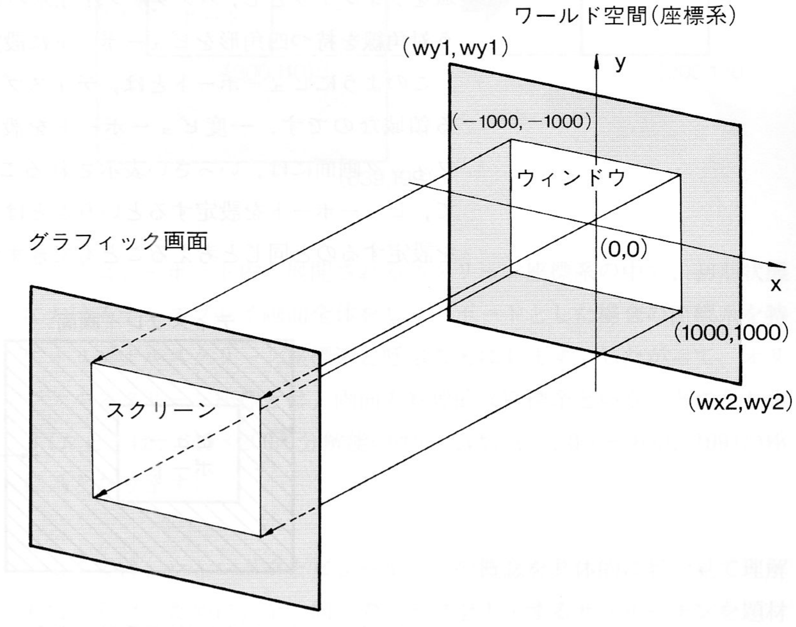
ビューポートを設定したとき，ビューポート内に存在する座標です．詳しくは，次節「ウィンドウとビューポート」の中で述べます．

3 ウィンドウとビューポート

1. ウィンドウ

ワールド座標系はもともと架空のものであり，我々が実際に見ることのできる座標はあくまでスクリーン座標なのです．

したがってユーザは，ワールド座標系の中から任意の領域を指定し，スクリーン座標上に表示領域として割り当てなければなりません．N_{88}-BASICでこの操作をしているのが後述する"WINDOW"命令であり，この指定されたワールド座標上の領域のことをウィンドウ(のぞき窓)と呼びます．次の図では，ワールド座標系の(−1000，−1000)から(1000，1000)の領域をウィンドウとしています．

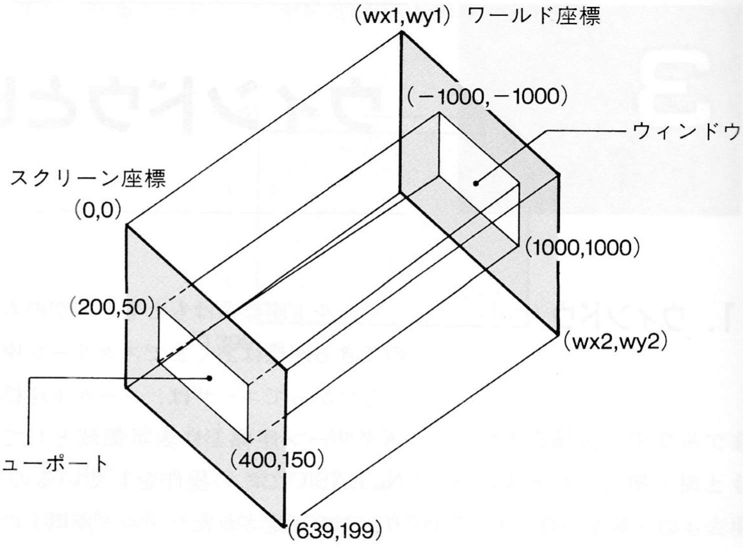


2. ビューポート

N_{88}-BASICではウィンドウとは別に"ビューポート"を設定することができます．

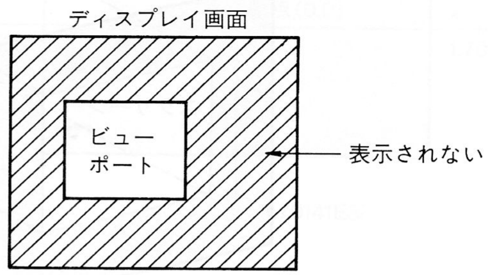
ウィンドウはワールド座標系における表示領域の設定であったわけですが，このビューポートは，ディスプレイ画面上における表示領域の設定です．この操作は，後述する"VIEW"命令によって行われます．

次の図は，ウィンドウとビューポートの関係です．



この図ではワールド座標系の(−1000，−1000)から(1000，1000)の領域をウィンドウとし，スクリーン座標系の(200，50)から(400，150)という対角線を持つ四角形をビューポートに設定しています．

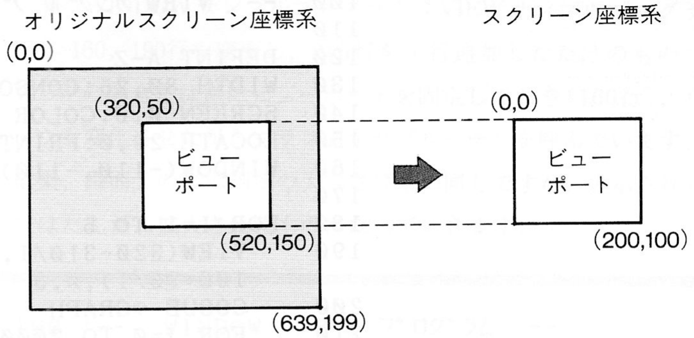
このようにビューポートとは，ディスプレイ画面上で見ることのできる領域なのです．一度ビューポートを設定すると，その外側のグラフィック画面には，いっさい表示されることはありません．したがって，ビューポートを設定するということはディスプレイ画面の使用領域を設定するのと同じと考えることもできます．

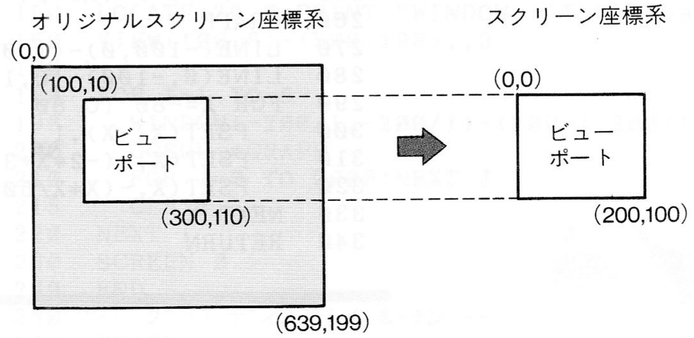


このビューポート内の座標系が，さきほど紹介したスクリーン座標系なのです．スクリーン座標系とはビューポートの中にのみ存在するもので，この座標系の各点はディスプレイ画面の１ドットに対応しています．これはあくまで物理的な意味の座標系であり，前記したワールド座標とは異なります．この座標系では，ビューポートの左上の頂点が必ず原点(0，0)に設定され，そこから画面上の各ドットを座標の１点として物理的に展開されます．

次の図はディスプレイ画面上の異なる位置に同じ大きさのビューポートを設定した例です．この場合，２つのビューポート内のスクリーン座

標系は全く等価となります．





このビューポート内に展開されるスクリーン座標系の中で，初期状態におけるディスプレイ画面全体をビューポートとした場合の座標系を特にオリジナルスクリーン座標系と呼ぶことにします．したがって，オリジナルスクリーン座標とは，画面の物理的な座標系ということになります(たとえば，640×200の分解能の画面では，(0，0)－(639，199)の座標系となります)．

3. サンプルプログラム(ウィンドウとビューポートの使い方)

ここでは，ウィンドウとビューポートの概念を具体的に目で見て理解していただくために，全く同一のグラフを表示するサブルーチンを題材に説明します．これは4象限の直交座標に "Y＝X"，"Y＝－2＊X＋15"，"Y＝X＊X/50－30" の関数をグラフ表示するものです．

(1)　ビューポートの概念を取り入れたサンプルプログラム

次ページのVIEWサンプルプログラムは，ウィンドウを固定しておき(160行)ビューポートの大きさを変化させながら，さきほどのグラフ表示サブルーチン(270行以降)を呼んでいます．実行結果は，グラフそのものの形状および情報はそのままで，表示される領域の大きさだけが変化しています．

```
100 '--  VIEW ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ  --
110 '
120  DEFINT A-Z
130  WIDTH 80,25:CONSOLE 0,25,0,1
140  SCREEN 0,0:COLOR 7,0:CLS 3
150  LOCATE 29,0:PRINT "VIEW ﾃﾞﾓﾝｽﾄﾚｰｼｮﾝ"
160  WINDOW(-110,-110)-(110,110)
170 '
180  FOR I=1 TO 5
190    VIEW(320-310/I,100-90/I)-(320+310/I,
       100+90/I),0,3
200    GOSUB *GRAPH
210    FOR J=0 TO 2000:NEXT J
220  NEXT I
230  SCREEN 0
240  END
250 '-- ｸﾞﾗﾌ ﾃﾞｨｽﾌﾟﾚｲ ﾙｰﾁﾝ --
260 *GRAPH
270  LINE(-100,0)-(100,0),7
280  LINE(0,-100)-(0,100),7
290  FOR X=-80 TO 80
300    PSET(X,-X),1                ' Y=X
310    PSET(X,-(-2*X+30)),2        ' Y=-2*X+15
320    PSET(X,-(X*X/50-30)),4 ' Y=X*X/50-30
330  NEXT X
340  RETURN
```

(2) ウィンドウの概念を取り入れたサンプルプログラム

下のWINDOWサンプルプログラムはVIEWサンプルプログラムの150，160，190行を変更し，215行を1行追加しただけのものです．VIEWの場合とは逆にビューポートを固定しておき(160行)，ウィンドウを変化させながらグラフ表示サブルーチンを呼んでいます．その結果，画面上の表示領域の大きさは常に同じですが，表示されるグラフはだんだん拡大されたものになっていきます．

```
100 '--  WINDOW ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ  --
110 '
120  DEFINT A-Z
130  WIDTH 80,25:CONSOLE 0,25,0,1
140  SCREEN 0,0:COLOR 7,0:CLS 3
150  LOCATE 29,0:PRINT "WINDOW ﾃﾞﾓﾝｽﾄﾚｰｼｮﾝ"
160  VIEW(100,8)-(500,198),,3
170 '
180  FOR I=1 TO 5
190    WINDOW(-200/I,-200/I)-(200/I,200/I)
200    GOSUB *GRAPH
210    FOR J=0 TO 2000:NEXT J
215    CLS 2
220  NEXT I
230  SCREEN 0
240  END
250 '-- ｸﾞﾗﾌ ﾃﾞｨｽﾌﾟﾚｲ ﾙｰﾁﾝ --
260 *GRAPH
270  LINE(-100,0)-(100,0),7
280  LINE(0,-100)-(0,100),7
290  FOR X=-80 TO 80
300    PSET(X,-X),1              ' Y=X
310    PSET(X,-(-2*X+30)),2      ' Y=-2*X+15
320    PSET(X,-(X*X/50-30)),4    ' Y=X*X/50-30
330  NEXT X
340  RETURN
```

このようにウィンドウとビューポートを適当に変化させることによって，同じ表示のサブルーチンであっても，画面上の任意の領域に拡大，縮小した図形が表示できますから，グラフのアレンジ，画面上でのレイアウトなど容易に行うことができます．

4 各座標系の指定方法

1. 指定方法の概略

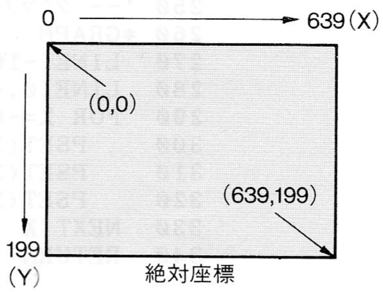
これまでいろいろな座標系を説明しましたが，ここではそれらの座標系に対して，種々のグラフィック命令を使う場合の"座標の指定の仕方"を説明します．

最も一般的な座標の指定は座標の絶対値によるもので，座標指定は普通どの座標系(キャラクタ座標系，スクリーン座標系，ワールド座標系)に対してもこの指定により行います．ただし，ワールド座標系に対しての座標の指定には，座標の相対値による方法も許されています．

次に，この２つの指定の仕方について説明します(640×200モードの場合)．

2. 絶対値による座標の指定

絶対値による座標の指定とは，グラフィックを描く位置を，座標系のX軸とY軸の値より行う方法です．



絶対座標

これは最も一般的な指定法で，BASICが持つすべての座標系に対して有効です．２章以降で解説するグラフィック命令などにおいて，座標指定する場合，普通はこの方法により行います．たとえば，PSET (X, Y)という命令はワールド座標系内の１点にドットをセットする命令ですが，これは次のような書式で書くことができます．

たとえば，

PSET (100, 100) ↵

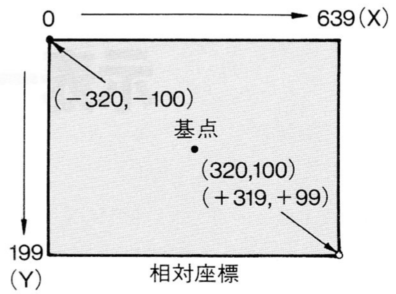
と入力します．この場合，ワールド座標の(100, 100)にドットがセットされます．その他の命令の場合も，絶対値による座標の指定ではすべてこのように(X, Y)という座標軸の値で指定します．

3. 相対値による座標の指定

ワールド座標系に描画を行うとき，絶対値による座標の指定とともに相対値による座標の指定も許されています．

相対値による座標の指定とは，座標系内に前もって設定した基点にX，Y方向の相対的な移動量を与えて，描画の座標を決める方法です．

たとえば，次のような座標系があり，基点が(320，100)に設定されていたとすると，左上の頂点は(−320，−100)であり，右下の頂点は(+319，+99)となるわけです．



この指定法の優れている点は，相対移動量を表す(X, Y)は常に同じ値であっても，基点の位置を動かすことによって違った位置を表現することができることです．

BASICはこの基点に当たるものとして，Last referenced Point (以後"LP"と呼びます)を持っています．このLPは最後に参照された座標という意味で，グラフィック命令で座標を与えた場合，最後に指定された座標を値として持ちます．

たとえば，POINT (X, Y)⏎ という命令が実行された場合，LPは(X, Y)という座標に設定されることになります．

グラフィック命令で相対値による座標を使って座標指定を行う場合は，次のような書式になります．

```
PSET STEP (X, Y)
```

これはドットをセットする命令の例ですが，その他の場合も同様で，座標を表す(X, Y)の前に"STEP"を付けて，(X, Y)が相対値であることを表します．

グラフィック命令で相対値による座標の指定が許されているのは次に示す命令です．

PSET	PRESET
LINE	PAINT
CIRCLE	POINT(ステートメントおよび関数)
GET@(ただし制約あり)	

第10章

カラー表示

N88-BASICは，フルカラーグラフィックを実現しています．表示色はパレットを使用し，8つの色を表示できます．また拡張命令を使用すれば，アナログパレットが使用できます．このときには，表示できる色は512色より8色を任意に選ぶことができます．
この章ではこのN88-BASICのカラー機能について，その発色の仕組みから各画面モードのときの注意事項までを解説していきます．

1 テキスト画面のカラー

1. カラーモードへの切り替え

テキスト画面は，普通はモノクロモードになっていますが，CONSOLE ,,,1 ⏎ を実行することによって，テキスト画面のカラー表示が可能となります．

このことは，0〜4のすべての画面モードの場合にあてはまります．

2. カラー指定を行う命令

CONSOLE ,,,1 ⏎ を実行し，カラーモードに切り替えたのち，実際にカラー指定を行うために使う命令としては，COLOR と COLOR@の2種があります．

COLOR文は，第1パラメータの数値に従って，以降に表示する文字の色を決定します．パラメータと表示色の関係は以下のとおりです．

0＝黒，1＝青，2＝赤，3＝紫，4＝緑，5＝水色，
6＝黄色，7＝白

COLOR@文は，範囲指定をしたうえで色指定を行い，その範囲にすでに表示されている文字の色を，色指定に従って変えます．色指定に使う数値と表示色の関係は，COLOR文の場合と全く同じです．

3. COLOR文の使用例

```
CONSOLE ,,,1 ⏎
COLOR 1 ⏎
```

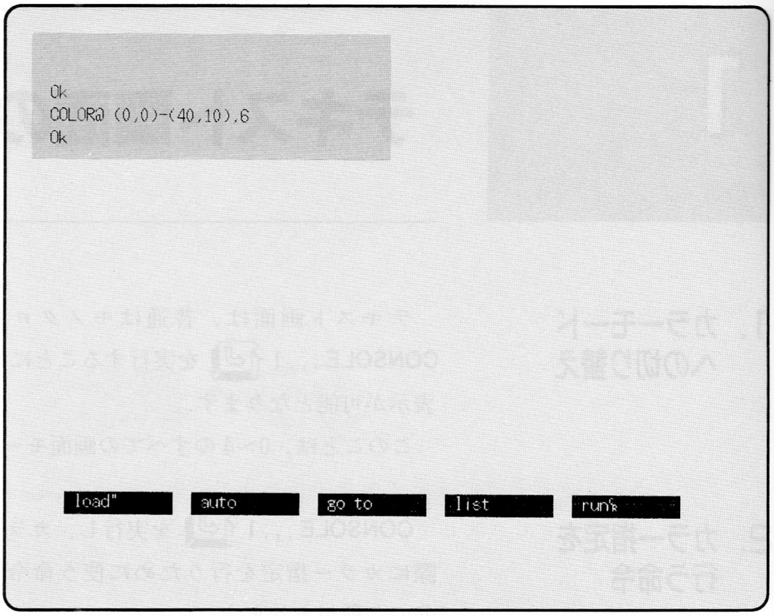
以上をダイレクトモードで実行したとすると，"Ok"も青で表示されます．カーソルはもとの色で表示されます．［画面消去］を押した時点でカーソルも青で表示されます．以後のキーボードからの入力やプログラムの実行結果も，すべて青で表示されます．

4. COLOR@文の使用例

表示桁数80×25の場合，

```
CONSOLE ,,,1 ⏎
COLOR@(0,0)−(40,10),6 ⏎
```

をダイレクトモードで実行したとすると，次ページの画面に［網掛け］で示した範囲にすでに表示されていた文字の色は，瞬時に黄色に変わります．


Ok
COLOR@ (0,0)-(40,10),6
Ok
load"
auto
go to
list
run

2 グラフィック画面のカラー

1. カラーグラフィックが可能な画面モード

グラフィック表示が白黒モードのときは，グラフィックのカラー表示は行えません．

グラフィック表示 ＼ 解像度	白黒モード	カラーモード
ヨコ　タテ 640 × 200	SCREEN 1 (N_{88}-BASIC)	SCREEN 0 (N_{88}-BASIC) SCREEN 3 (N_{88}-日本語BASIC)
ヨコ　タテ 640 × 400	SCREEN 2 (N_{88}-BASIC) SCREEN 4 (N_{88}-日本語BASIC)	

2. ドット単位で8色中1色の指定が可能

テキスト画面のカラーではキャラクタ単位で色指定を行いましたが，グラフィック画面のカラーではドット単位で８色中１色を指定することができます．

使用できる色は，テキスト画面のカラーと同じ８色です．

このことにより，隣り合ったドットであっても完全に別の色で塗り分けることができます．

3. CIRCLE文，LINE文における使用例

N_{88}-BASICのグラフィック命令のほとんどすべてが，カラーグラフィックを実現できます．たとえば，

```
CIRCLE(320,100),50,4 ⏎
```

をダイレクトモードで実行すれば，画面中央に緑の円が描けます．また，

```
LINE(0,0)−(200,150),2,BF ⏎
```

をダイレクトモードで実行すれば，画面の左側上２/３の部分あたりに，赤で塗りつぶされた四角形が描かれます．

3 パレット

1. パレットの機能

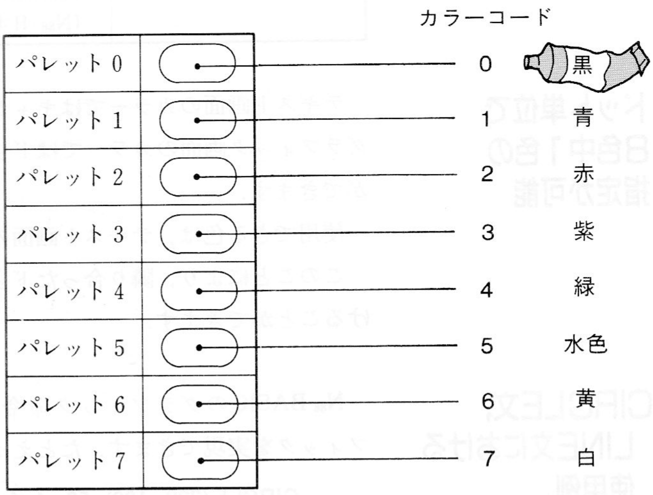
N_{88}-BASICは，カラー表示を行うために，パレットという考え方を導入しています．

パレットとは文字どおり絵皿のことです．N_{88}-BASICではパレット0～パレット7までの8個のパレットを持っています．ユーザはこれらのパレットに好みの色を設定することができます．

いわば，パレットは間接色指定機能ともいうべきもので，これによりカラー表示の融通性や汎用性を高めています．

2. パレットの初期設定

N_{88}-BASICを起動したときのパレットとその設定色を次に示します．パレット0に黒，パレット1に青……と設定してあるのがわかります．



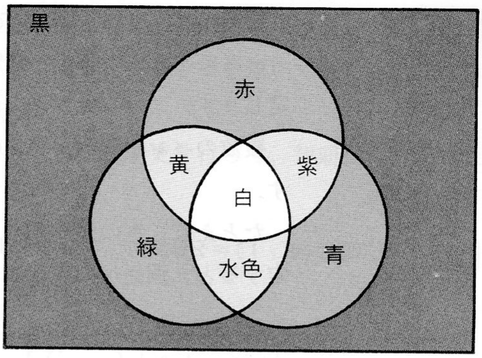
なお，右側にある数字はカラーコードといって，実際の表示色に対応します．

3. カラーコードと表示色の関係

カラーコードと表示色の関係を理解するには，カラーディスプレイにおける発色の原理を知るのが近道です．

一般にカラーディスプレイは，光の三原色(赤，青，緑)をもとに，その組み合わせによって多彩な色彩の表示を行っています．

すなわち，三原色が単独に表示されれば各原色の色そのものであり，任意の複数の組み合わせで中間色および白が得られます．三原色ともに表示されなければ，そのドットは黒となります．



これを計算機上で行うには，画面上の各ドットに対し，最低3ビット（赤用，青用，緑用）を用意し，各ビットの0，1に従って，発光ON/OFFの制御をすればいい，ということになります．

この原理により求められたものが，カラーコードです．

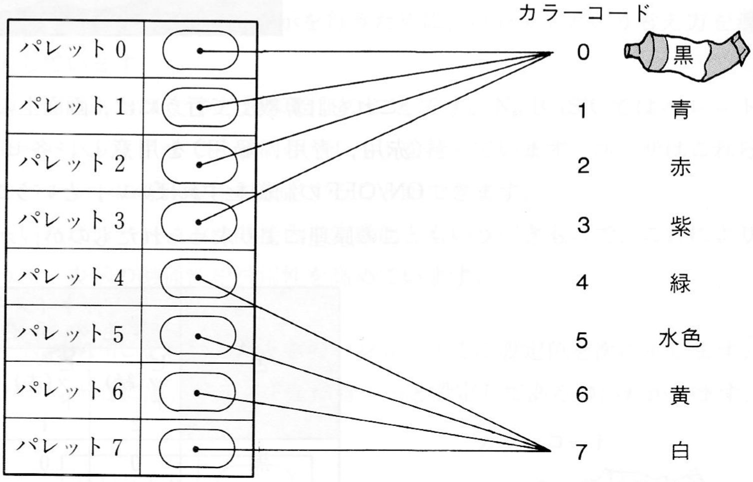
色	ビットイメージ 色の組合せ ビット2(緑)	ビット1(赤)	ビット0(青)	カラーコード
黒	0	0	0	0
青	0	0	1	1
赤	0	1	0	2
紫	0	1	1	3
緑	1	0	0	4
水色	1	0	1	5
黄色	1	1	0	6
白	1	1	1	7

4. パレットの設定

実際にパレットを設定するにはCOLOR＝文を用います．第１パラメータにパレット番号を指定し，第２パラメータにはカラーコードを指定します．

複数のパレットへ，同一のカラーコードをセットすることもできます．

たとえば，次のようにパレット0～3を黒に，パレット4～7を白に設定すると，画面への表示は白と黒の２色になります．



5. パレットによる表示色の変更

パレットは，すでに表示したグラフィックの色の変更にも使えます．

これをプログラムを通して見てみます．次のサンプルプログラムを作成，実行してください．

```
100 SCREEN 0
110 WINDOW(0,0)－(639,199)            画面の初期化
120 VIEW(0,0)－(639,199)
130 COLOR＝(1,1)  ──────→ パレット１に青を設定
140 LINE(0,0)－(639,199),1            始点(0,0)から終点(639,199)までパレット１の色で線を描く
150 COLOR＝(0,0)  ──────→ パレットをもとに戻す
160 END
 RUN
```

画面に青い線が描けたはずです．ここでパレット１の色を赤に変更します．

```
COLOR＝(1,2)
```

画面の線が赤い表示に変わったはずです．このようにパレットを利用すれば，グラフィックに色の変化を与えることができます．なお，このプログラムでは線を引くだけでしたが，実際にはグラフィックで描いた

小動物の目の色を変えるなど幅広い応用が考えられます．パレットを使ううえでの注意として，パレットを用いて色を変更したら，必ずもとに戻してください．

なお，パレットの設定変更を行って，初期設定に戻したい場合は，次のようにしてください．

```
FOR I=0 to 7:COLOR=(I,I):NEXT I ⏎
```

6. テキスト画面でのパレットの使用

グラフィック表示モードがカラーでない場合(SCREEN 1,2,4)においては，CONSOLE,,,1 ⏎ を実行して，テキスト画面をカラーモードにすると，テキスト画面でもパレット機能を利用することができます．

4 アナログパレット

1. アナログパレットとは

N88-BASIC/N88-日本語BASICでは，NEW CMDを実行することによって拡張命令が追加されます．追加された拡張命令の中のCMD PALを用いてパレット指定することにより，ユーザは512色のうちから８色を選ぶことができます．これにより，"明るい赤"や"淡い赤"などの中間色を表示することができます．

ただし，この場合は，アナログRGBディスプレイが本体に接続されていなければ意味を持ちません．アナログRGBディスプレイが接続されている場合のみ，基本色以外の色を表示することができます．

ディジタルRGBディスプレイ（８ピンコネクタインタフェースを持つディスプレイ）が接続されている場合，基本色以外の色がCMD PAL文でパレットに設定されても，表示される色は基本色となります．

2. アナログパレットの初期設定

N88-BASIC/N88-日本語BASICにおいて，CMD PAL文をパラメータなしで実行した直後のパレットの状態は次のようになります．

	カラーコード	
パレット０	&O000	黒
パレット１	&O007	青
パレット２	&O070	赤
パレット３	&O077	紫
パレット４	&O700	緑
パレット５	&O707	水色
パレット６	&O770	黄
パレット７	&O777	白



アナログカラーコードはカラーコードと同じ機能を持ちます．512色中の８色を使ったときではアナログカラーコードと呼び，カラーコードと区別しています．

3. アナログカラーコードと表示色の関係

通常の0〜7を示すカラーコードでは，赤・青・緑の発光のON/OFFで８種類の色を表現していました．

アナログカラーコードでは，赤・青・緑の各原色の発光に８段階（３ビットデータの指定）の輝度を与えています．これにより，8×8×8＝512通りの色の組み合わせを実現しています．なお，実際の表現色とアナログカラーコードの対応は次のとおりです．

色	色の組み合わせ（輝度）			
	緑	赤	青	
黒	0	0	0	&O000
1/7 暗い青	0	0	1	&O001
2/7 暗い青	0	0	2	&O002
3/7 暗い青	0	0	3	&O003
：	：	：	：	：
7/7 明るい青	7	0	0	&O007
：	：	：	：	：
赤	0	7	0	&O070
：	：	：	：	：
緑	7	0	0	&O700
白	7	7	7	&O777

4. アナログパレットの設定方法

パレットを設定する方法は，CMD PAL文の第１パラメータにパレット番号を指定し，第２パラメータにはアナログカラーコードを指定します．また，"&O000〜&O777"（10進数で０〜511）の範囲でアナログカラーコードを指定します．アナログカラーコードの指定方法は，CMD PAL命令を参照してください．

CMD UNLINKを実行すると拡張命令が取り去られ，CMD PALを使用することができなくなります．ただし，拡張命令が取り去られる前に設定したパレットは，拡張命令が取り去られたあとでも使用することはできます．

5. サンプルプログラム（アナログパレットの使い方）

次のサンプルプログラムは８色のカラーを画面に表示し，その中の１つのパレットを指定し（270行），青の輝度，赤の輝度，緑の輝度を指定（300行から410行）することによって，指定したパレットの色を変えます．パレットナンバーを指定するところで"−1"を入力すると，パレットを初期状態に戻してから終了します．パレットを変更した場合，必ず

このようにパレットを初期状態に戻すよう心がけてください．

また，このプログラムを実行する場合は，本体にアナログRGBディスプレイを接続することが必要です．

```
100 '-- ｱﾅﾛｸﾞ ﾊﾟﾚｯﾄ ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ --
110 '
120 WIDTH 80,25:CONSOLE 0,25,0,1:SCREEN 0,0
130 NEW CMD:CLS 3:CMD PAL
140 LOCATE 20,1
150 PRINT "ｱﾅﾛｸﾞ ﾊﾟﾚｯﾄ ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ"
160 FOR X=40 TO 544 STEP 72
170   LINE(X,50)-(X+71,138),INT(X/70),BF
180 NEXT X
190 FOR I=0 TO 7
200   LOCATE 8+CX,5:PRINT I
210   CX=CX+9
220 NEXT I
230 '
240 *LOOP
250  DEC=0
260  LOCATE 2,19
270  INPUT "ﾊﾟﾚｯﾄ No. (0-7 or -1=end) ";A
280   IF A=-1 THEN *EXIT
290   IF A<-1 OR A>7 THEN LOCATE 32,19:
                          PRINT SPACE$(48):
                          GOTO 260
300  LOCATE 45,19
310  INPUT "GREEN INTENSITY ( 0 - 7 ) ";B
320   B$=MID$(STR$(B),2)
330   IF B<0 OR B>7 THEN LOCATE 72,19:
                         PRINT SPACE$(8):
                         GOTO 300
340  LOCATE 2,21
350  INPUT "RED INTENSITY ( 0 - 7 ) ";C
360   C$=MID$(STR$(C),2)
370   IF C<0 OR C>7 THEN LOCATE 28,21:
                         PRINT SPACE$(52):
                         GOTO 340
380  LOCATE 45,21
390  INPUT "BULE INTENSITY ( 0 - 7 ) ";D
400   D$=MID$(STR$(D),2)
410   IF D<0 OR D>7 THEN LOCATE 71,21:
                         PRINT SPACE$(9):
                         GOTO 380
420  LEVEL$=B$+C$+D$
430  FOR I=1 TO 3
440    NUM=VAL(LEFT$(LEVEL$,1))
450    LEVEL$=MID$(LEVEL$,2)
460    DEC=8*DEC+NUM
470  NEXT
480  CMD PAL A,DEC
490  CONSOLE 19,6:CLS
500 GOTO *LOOP
510 '
520 *EXIT
530  CMD PAL
540  CONSOLE 0,25
550 END
```

5 ラインスタイルとタイリング

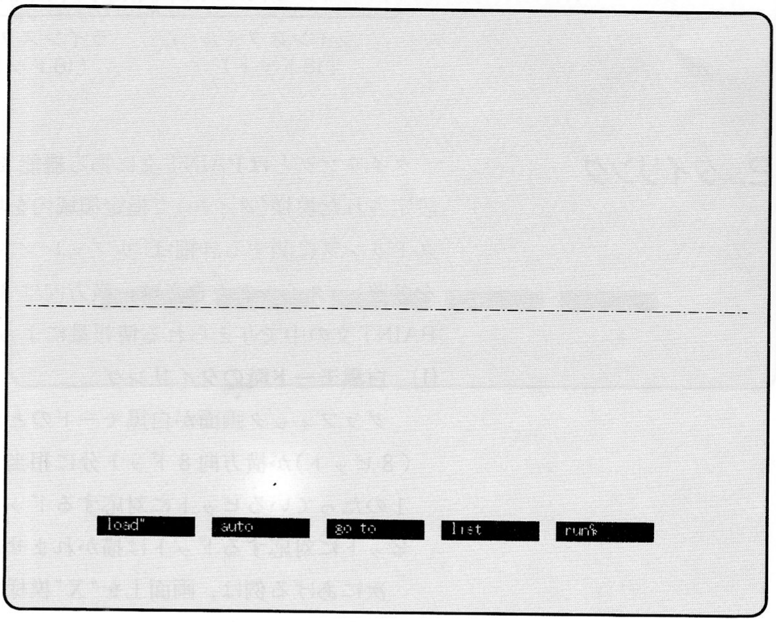
1. ラインスタイル

ラインスタイルとはLINE文にある機能で，引く線の種類を設定します．これを設定するためには，16ビットで表現可能な数(&H0～&HFFFF)の範囲の値を用います．ラインスタイルは16進数で指定しなくてもかまいませんが，16進数で表現すると便利です(詳細は，「リファレンスマニュアル」のLINE文の項を参照してください)．

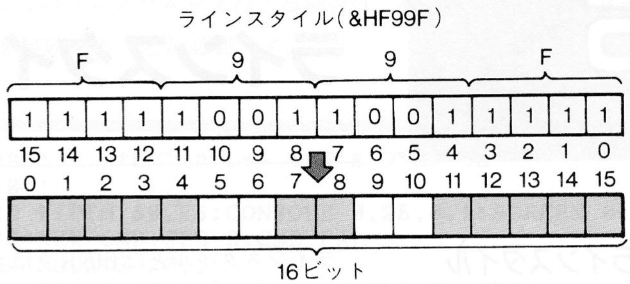
たとえば，

```
LINE(0,100)-(639,100),7,,&HF99F ⏎
```

を実行すると，下図のような点線が引かれます．

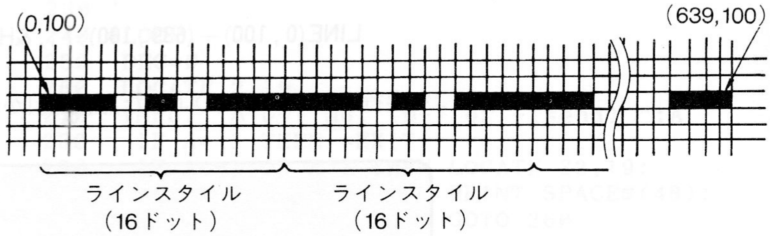


ラインスタイルを **&HF99F** と設定すると，次のようになります．



※　同じ番号のビットが対応します．

つまり，このラインスタイルが(0，100)～(639，100)まで繰り返されるため，図のような点線が現れるのです．



2. タイリング

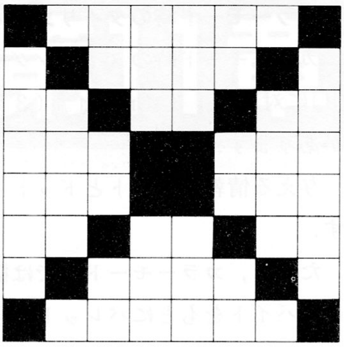
タイリングとはPAINT文にある機能で，タイルストリングによって設定された模様(タイル)で指定領域内を埋めるときに使います(タイルストリングに関する詳細は「リファレンスマニュアル」のPAINT文の項を参照)．タイルの大きさは，横方向は8ドット，縦方向の大きさは，PAINT文の中で与えられる情報量によって決まります．

(1)　白黒モード時のタイリング

グラフィック画面が白黒モードのときのタイリングでは，1バイト(8ビット)が横方向8ドット分に相当します．そして，2進数表現で1のたっているビットに対応するドットは描かれ，1のたっていないビットに対応するドットは描かれません．

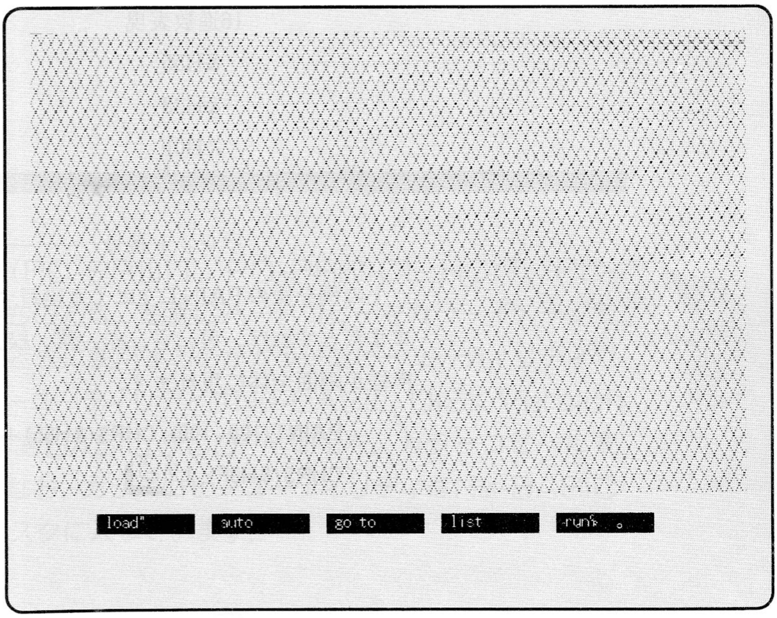
次にあげる例は，画面上を"X"模様でぬりつぶします．下記のとおり入力してください．

```
PAINT(320，100)，CHR$(&H81)＋CHR$(&H42)＋CHR$(&H24)
＋CHR$(&H18)＋CHR$(&H18)＋CHR$(&H24)＋CHR$(&H42)
＋CHR$(&H81)，1 ⏎
```



(0,0)	CHR$(&H81)	タイルバイト 1
(0,1)	CHR$(&H42)	タイルバイト 2
(0,2)	CHR$(&H24)	タイルバイト 3
(0,3)	CHR$(&H18)	タイルバイト 4
(0,4)	CHR$(&H18)	タイルバイト 5
(0,5)	CHR$(&H24)	タイルバイト 6
(0,6)	CHR$(&H42)	タイルバイト 7
(0,7)	CHR$(&H81)	タイルバイト 8

なお，このときの画面は次のようになります．



(2) カラーモード時のタイリング

カラーモード時のタイリングでは，縦方向のドット（×ヨコ8ドット）に対して，3×nバイト（3×n×8ビット）の情報を与える必要があります．

与える情報のビットとドットの関係は白黒モードのときと同様です．

ただし，カラーモード時では横8ドット分として与えられた場合，×nバイトをもとにパレット番号を算出し，そのとき，それぞれのパレットにセットされているカラーコードに従って表示を行います．

例 CHR$(&H55)＋CHR$(&H33)＋CHR$(&H0F)

16進数表現	2進数表現
&H55	01010101
&H33	00110011
&H0F	00001111
	↓↓↓↓↓↓↓↓
パレット番号 ──────→	01234567

上の例を次のPAINT文により実行すると，パレットが初期設定のままだとすれば，黒，青，赤，紫，緑，水色，黄色，白の細かい縦縞模様が画面全体に現れます．

```
PAINT(100,100),CHR$(&H55)+CHR$(&H33)+
CHR$(&H0F),1 ⏎
```

第11章

カラーコピーユーティリティ

カラープリンタ(PC-PR101T，PC-PR201CL・HC・T，PC-PR406)をお持ちの方は，画面のハードコピーをカラーイメージで行えます．このときに使用するのがカラーコピーユーティリティです．カラーコピーユーティリティはN88-BASIC DISK VersionおよびN88-日本語BASICのCOPY命令を，カラーコピーに変えるものです．なお，カラーコピーユーティリティはN88-BASICのシステムディスクに入っています．

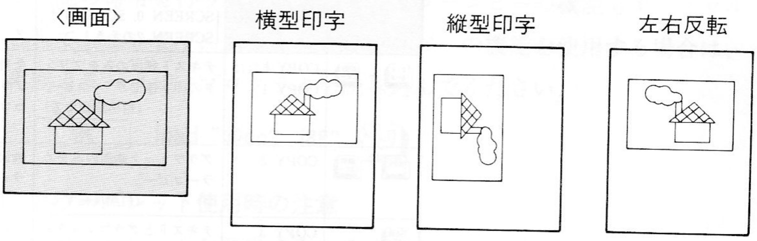
1. ユーティリティの構成

N_{88}-BASICシステムディスクには，次の4本のカラーコピー用プログラムが入っています．

対応機種	BASICプログラムのファイル名	印字形式	使われるメモリの大きさ
ドットインパクトプリンタ PC-PR201CL, PC-PR201H2, PC-PR201V, PC-PR201HC	@cc2. j88	横型印字	約2.4Kバイト
熱転写プリンタ PC-PR101T, PC-PR101TL, PC-PR201TL, PC-PR201T, PC-PR406	@cc4v. j88	縦型印字	約2.5Kバイト
	@cc4h. j88	横型印字	約2.5Kバイト
	@cc4m. j88	左右反転	約2.5Kバイト

それぞれの印字形式によって，プリンタには次のように印字されます．

ただし，COPY 1(テキスト画面のみのカラーコピー)では，どの印字形式でも横型に印字されます．



A4サイズの用紙に横型印字を行うときは，左右の余白が少ないので，用紙のセット位置に気をつけないと用紙からはみ出してコピーしてしまいます．PC-PR201シリーズのプリンタでは大きい用紙を使うとよいでしょう．それ以外の用紙幅の狭いプリンタでは縦型印字にすると簡単にできます．

2. 操作手順

ユーティリティを起動するときは，次の要領で行ってください．

(1) N_{88}-BASIC DISK version，またはN_{88}-日本語BASICを起動します．

(2) 他の機械語プログラム(拡張命令など)も使う場合は，「5. メモリマップ」を先にお読みください．

(3) 本プログラムの入っているフロッピーディスクをドライブ1にセットし，お手持ちのプリンタの機種に合ったBASICプログラムを選んで実行します．

例　run "@cc2 . j88" ↵

機械語ファイルが読み込まれ，以後COPY CUT ↵ を入力する

まで，BASICで画面のカラーコピーコマンドが使えます．

なお，画面に"Out of memory"エラーが表示された場合は，「5. メモリマップ」を読んでから，正しい手順でもう1度読み込みなおしてください．

(4) カラーコピーコマンドが不要になったり，従来の白黒コピーを使用するときは

copy cut ⏎

を入力します．これで，プログラムが専有していた領域もフリーエリアとして解放されます．また，他の印字形式のカラーコピープログラムを使う場合も，COPY CUT ⏎ を入力してから，目的のプログラムを読み込んでください．

3. ユーティリティの機能

ユーティリティの起動により，COPY命令の機能は次のように変わります．

キー	コマンド	機能		
		N_{88}-BASIC使用時，N_{88}-日本語BASIC使用時 (SCREEN 0, SCREEN 1, SCREEN 2のとき)	N_{88}-日本語BASIC使用時 (SCREEN 3のとき)	N_{88}-日本語BASIC使用時 (SCREEN 4のとき)
コントロール + 画面コピー	COPYまたは COPY 1	テキスト画面のみをプリンタの印字体でカラーコピー．(白黒反転する)	全角文字を含むテキストをプリンタの印字体でモノクロコピー．(白黒反転する)	全角文字を含むテキストをプリンタの印字体でカラーコピー．(白黒反転する)
GRPH + 画面コピー	COPY 2	グラフィック画面のみをカラーコピー．(白黒反転する)	画面イメージをそのままカラーコピー．(白黒反転する)	画面イメージを，そのままモノクロコピー．(白黒反転する)
画面コピー	COPY 3	テキストとグラフィックの両画面をカラーコピー．(白黒反転する) テキスト画面もそのままのイメージでドット出力．		画面イメージを，そのままカラーコピー．(白黒反転する)
——	COPY 4	グラフィック画面のみをカラーコピー．(白黒反転しない)	画面イメージをそのままカラーコピー．(白黒反転しない)	画面イメージを，そのままモノクロコピー．(白黒反転しない)
——	COPY 5	テキストとグラフィックの両画面をカラーコピー．(白黒反転しない) テキスト画面もそのままのイメージでドット出力．		画面イメージを，そのままカラーコピー．(白黒反転しない)
——	COPY CUT	カラーコピーを無効にし，従来のCOPY命令(白黒)を有効にする．プログラムが専有していた領域を解放する．		

白黒反転する場合は，画面上の白が黒に，黒が白になって印字されます．白黒反転しない場合は，画面と同じ色で印字されます．

注意 各BASIC(カラーコピー)プログラム1210行目のREMを消すと，GRPH + 画面コピー はCOPY 4の機能に，画面コピー はCOPY 5の機能になります．たとえば"@cc2 . j88"の場合は次のような手順になります．

```
load "@cc2.j88"
list 1210
1210 REM POKE BTM.BODY,0
REMを削除して
1210 POKE BTM.BODY,0
run
```

4. 使用上の注意事項

カラーコピーユーティリティは，COPY命令の拡張を行うものです．COPYの拡張部分はメモリの空き領域に格納されます．ですから，メモリ領域を操作する命令や拡張命令(NEW CMD)を使うときは注意が必要です．

(1)　拡張命令のキャンセル

これらの拡張命令を消し去る(キャンセルする)ときは，必ずCOPY CUTを先に行ってから，CMD UNLINKを行ってください．

```
copy cut
cmd unlink
```

上記の命令を行うことによって，カラーコピーの機能もキャンセルされてしまいます．もう1度カラーコピーの機能を使用する場合は，再度カラーコピープログラムを読み込んでください．

例　load "@cc2.j88"

(2)　カラーパレット使用時の注意

カラーパレットを変更するときは，カラーコピープログラムを読み込んだ後，COLOR=(パレット番号，カラーコード)文あるいはCMD PAL文で設定してください．OUT文で変更したり，カラーコピープログラムを読み込む前に変更すると，正しくコピーされませんので，プログラムを読み込んだ後にもう1度カラーパレットを設定しなおしてください．

なお，アナログRGBディスプレイを接続して拡張命令(NEW CMD)を追加し，CMD PAL文で512色の中から任意のアナログカラーコードを設定してカラーコピーを行った場合，画面と同じ色合いにならないことがあります．

(3)　リバース・ブリンク表示している文字の印字

テキスト画面が白黒モードのとき，リバース表示している文字は，COPY 1では通常の表示として印字されます．ブリンク表示は常にノーマルで印字されます．

(4) **レフトマージン(左余白)の設定**

プリンタにあらかじめレフトマージンを設定しておくと，用紙の左側に空白を作ることができます．レフトマージンは

```
lprint chr$(27) ; "L" ; "nnn" ; ↵
```

(nnnは3桁の10進数でレフトマージン幅)

と指定します．レフトマージン幅を2にしたいときは"nnn"のところに"002"と入力します．

5. メモリマップ

ユーティリティのメモリへの格納は，空き領域(CLEAR命令参照)の上位アドレスより行われます．ですから，拡張命令(NEW CMD)および機械語プログラムを使うときは，それらを先に読み込んでおく必要があります．

ただし，メモリの空き領域が十分にないと，読み込むことができず"Out of memory"エラーが発生します．この原因は，メモリの上限が小さい，もしくはファイルバッファの数(BASIC起動時に"How many files(0～15)?"に入力した値)が大きいため，ユーティリティの載るスペースが不足しているのです．このような場合は，CLEAR文の第2パラメータを大きな値に設定しなおして，プログラムサイズ分の領域を確保してから，もう1度カラーコピープログラムを読み込んでください．

なお，カラーコピープログラムを読み込んだ後で，それより大きい番地に機械語プログラムを読み込んだり，CLEAR文を設定したりすることはできません．

次に例として，本機のN_{88}-BASIC V2モードで拡張命令を追加し，カラーコピープログラムも使用するときの手順を示します．

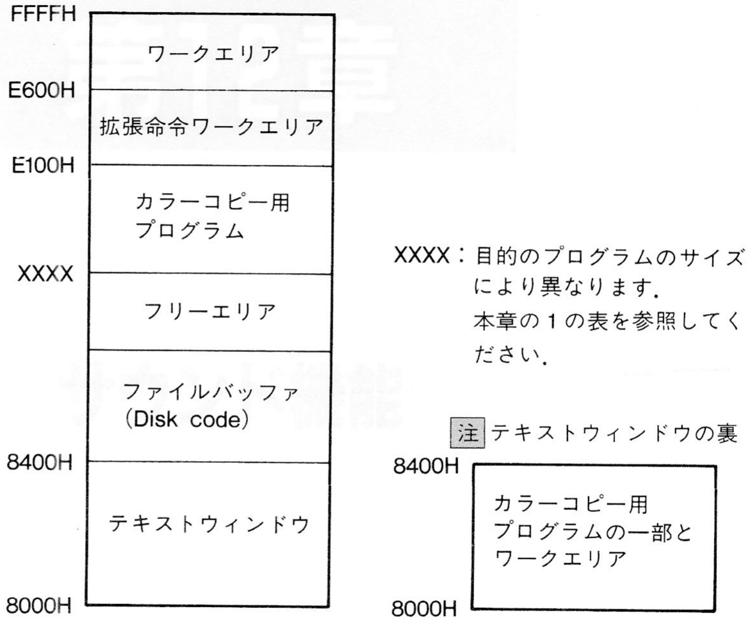
まず拡張命令を追加します．

```
NEW CMD ↵
```

続いて，目的のカラーコピープログラムを読み込みます．

```
run "@cc2 . j88" ↵
```

このとき，メモリマップは次の図のようになっています．



注　テキストウインドウの裏は，カラーコピー用プログラムの一部およびワークエリアとして使います.

カラーコピーの機能が不要になったらCOPY CUTを行ってください．プログラムが専有していた領域はフリーエリアとして解放されます．なお，上のメモリマップに示されている拡張命令を取り去るときには，必ずCOPY CUTを先に行ってからCMD UNLINKを行ってください．くれぐれもお忘れなく！

6. ユーティリティの転送

4本のカラーコピープログラムは次の表のように，それぞれBASICプログラム1本と機械語プログラム3本ずつから成っています.

したがって，カラーコピーユーティリティを他のフロッピーディスクに転送するときは，BASICプログラムだけでなく，機械語プログラムも転送する必要があります.

対応機種	印字方式	BASICプログラムのファイル名	機械語プログラムのファイル名		
ドットインパクトプリンタ PC-PR201CL, PC-PR201HC, PC-PR201H2, PC-PR201V	横型印字	@cc2. j88	@cc2*rel	@cc2*map	@cc2*bin
熱転写プリンタ PC-PR101T, PC-PR201T, PC-PR101TL, PC-PR406, PC-PR201TL	縦型印字	@cc4v. j88	@cc4v*rel	@cc4v*map	@cc4v*bin
	横型印字	@cc4h. j88	@cc4h*rel	@cc4h*map	@cc4h*bin
	左右反転	@cc4m. j88	@cc4m*rel	@cc4m*map	@cc4m*bin

ディスクユーティリティの「ファイル転送」("xfiles . j88")を使って必要なファイルをコピーしてください(第16章参照).

第12章

サウンド機能

本機にはシンセサイザIC（OPN）が内蔵されており，強力なサウンド機能を持っています．これにより，6重奏や効果音を出すことができ，さらにユーザ自身で音色を定義することもできます．

1 音の原理と音源の種類

コンピュータで音を制御するには，まず音の原理を知らなくてはなりません．

私たちが日常聞いている音は，空気を伝わって聞こえてきます．

それでは音の出ている場所，つまり音源ではどのようになっているのでしょうか？

今ここに1つのスピーカがあるとします．このスピーカから音が出ていれば，そのスピーカが音源ですね．スピーカから音が出るのは，スピーカのコーン紙が電気信号によって振動して空気をふるわせているからです．

でも，ちょっと考えてください．単に空気が振動していればよいのでしょうか？ それではただの雑音と変わりません．

音，いいかえれば音楽であるためには，いくつかの要素が必要です．

まず音程です．これは学校の音楽の時間で習った，あのドレミなどで表される音の高さです．

次に，音符や休符で表される音の長さです．

3つめに音の強弱，そして音そのものの特徴を表す音色の4つがなければ，音楽として成り立ちません．

これらの4つの要素を制御する，特に音の強弱や音色を制御するのはとても大変です．

しかし，この4つの要素をすべて制御することを可能にしたのが，後で述べるFM音源なのです．

ところで，本機は一体いくつの音源を持っていると思いますか？ 実は，BEEP音源，SSG音源，FM音源の3つでBEEP音から6重奏の音楽まで作っているのです．

それでは，それぞれの音源を簡単に説明しましょう．

1. BEEP音源

BEEP音源は，あらかじめ設定された周波数で音を出します。プログラムの最中にエラーを出したときに鳴る，あのピーッという音がそうです。

2. SSG音源

SSG音源は，シンセサイザICのSSG音源部から音を出しています。SSG音源は，後述のFM音源に比べて簡単に音を出すことができます。特にリズムや効果音などに向いているでしょう。

3. FM音源

FM音源は，シンセサイザICのFM音源部から音を出しています。FM音源にはオペレータというシステムがあります．このオペレータを組み合わせることによって様々な音色(たとえば，ピアノの音や犬の鳴き声など)を作り出すことができます．

2 音を出すときに使うBASICの命令

1. BEEP

N_{88}-BASICおよびN_{88}-日本語BASICでは，BEEP, BEEP 1, BEEP 0の3つの命令でBEEP音を制御しています(BEEP文の使用方法は「リファレンスマニュアル」"BEEP"を参照してください).

2. 拡張命令

拡張命令にはサウンド機能を強化した命令がたくさんあります．特にCMD PLAYやCMD VOICEは，FM音源とSSG音源による美しい6重奏や，自分で音色を作ったりできます．

また，CMD SOUNDはインタフェースボードを介して楽器を接続することにより楽器を制御することもできます．

3 実際にサウンド機能を体験してみよう

ここではプログラムを使って約50種類の音色をそれぞれ聞いたり，ちょっと手間がかかりますが，自分でオルガンの音を作ったり，サウンドユーティリティプログラムを実際に使って本機が持っている様々なサウンド機能を体験してみましょう．

では，さっそくみなさんをサウンドの世界へご案内します．

1．サウンド機能を使うには

ディスプレイ，本体の順に電源をONにしてN_{88}-BASIC V2またはN_{88}-日本語BASICを起動させたあと

```
NEW CMD ↵
```

と入力してください．"Ok"という表示が出れば，拡張命令が使えるようになります．

2．音を出す

下に示してあるのが，約50種の音色を出すプログラムです．実際に入力し，実行してみましょう(行の最後で ↵ を押し忘れないでください)．

```
10 FOR K=0 TO 61 ↵
20 CMD PLAY "@=K;V13O5L4CEG" ↵
30 NEXT K ↵
RUN ↵
```

いかがでしたか？ いろいろな音色を持っていますね．今聞いてもらったのは，あらかじめ本機に設定された楽器や効果音の音色です．他にもユーザ自身で別の音色を定義することもできます．そのときに使うのが，CMD VOICEというステートメントです(詳しくは「リファレンスマニュアル」のCMD VOICEを参照してください)．

3．CMD VOICEを使って音を作る

音作りで最も重要なのが，音の時間的変化です．これをエンベロープ(Envelop)といいます．そして，作りたい楽器の特徴をよく理解することも大切なことです．

たとえばピアノであれば，鍵盤を押したとき鋭い音の立ち上がりがあり，そのあと続けて押していれば，徐々に音が消えていくエンベロープを持っています．また，倍音の構成も立ち上がりのときに多く，時間が

経つにつれて倍音の数は少なくなり，一定の倍音構成に近づいていきます．以上のような特徴をつかんで音作りを進めていきます．

エンベロープを人工的に作る機能を，EG（Envelop Generator）といいます．そのパラメータとしてAR（Attack Rate），DR(Decay Rate)，SR(Sustain Rate)，RR(Release Rate)，SL(Sustain Level)などがあります．

これらを設定していくためには配列変数を使います．この配列変数の各値はそれぞれ特別な意味を持っており，それらの値をCMD VOICE命令で音色として設定します．配列変数の意味は，「リファレンスマニュアル」のCMD VOICE文を参照してください．

まず，配列変数を使うためにDIM文で宣言します．

```
DIM A%(4, 9) ⏎
```

FM音源で音を出すためには，オペレータが重要な役割を果たしています．このオペレータは4つあり，それぞれをいろいろと組み合わせることによって，出る音が全く違ってきます．その組み合わせをアルゴリズム(Algorithm)といい，8つのパターンがあります．

4つあるオペレータはモジュレータとキャリアに分かれ，これはアルゴリズムによってその対応が変わります．

キャリアとモジュレータのパラメータの効果は下図のようになっています．

項　　目	関与するパラメータ	設定値による音の変化 値 MIN ↔ MAX
キャリアの出力レベル	各オペレータのOutput Level (AR, DR, SR, RR, SL)	音量大↔音量小 (注：Level は0が最大となります)
モジュレータの出力レベル		丸い音色↔明るい音色
モジュレータのフィードバックレベル	Feed back	普通の音色↔鋭い音色ノイズ
キャリアの周波数	各オペレータのDetune, Multiple	ピッチ低↔ピッチ高
モジュレータの周波数		近い倍音↔離れた倍音

8つのアルゴリズムは大きく3つに分けられます．

(1)　キャリア数が1つのアルゴリズム(番号0〜3)

1つのキャリアに対して残りのすべてのオペレータがモジュレータとなるわけですから，強力に明るい音になります．またこのアルゴリズムは，極端な音色変化だけでなく，モジュレータの出力レベルを控え気味にすれば，複雑な波形を持つ微妙な音色を作ることも可能であり，ソロ楽器向きといえます．

(2)　キャリア数が2つのアルゴリズム(番号4)

キャリア数が2つのアルゴリズムは，音作りの結果を想像しやすいわりに幅広い音色が作れるオールマイティなパターンです．2つのキャリアのピッチをずらしてコーラス効果を出したり，一方で基本的な音色を作り，他方では味付け的な効果(たとえば，息の音)を出したりすることも可能です．

(3)　キャリア数が3つか4つのアルゴリズム(番号5～7)

厚みのある落ち着いた音色を作るのに向いています．すべてのキャリアのピッチを少しずつずらしてやると，多数の楽器を同時に弾いているようなコーラス効果が作れます．また，アルゴリズム7のように4つのオペレータの出力をすべて加え合わせるパターンは，オルガン系の音色を作るのに最適です．

それでは，これからエレクトリックオルガンの音を作ることにしましょう．

そこでまず初めに，これら4つのオペレータをONにして使用可能(音を出せる状態)にするために，次のようにしてください．

```
A%(0, 1)=15 ⏎
```

次にアルゴリズムを選ぶわけですが，前にも述べたように，オルガン系はアルゴリズム7がよいのですが，ノイズを付けてボリューム感のある音を作るためにアルゴリズム5を選びます．アルゴリズムの設定は配列(0, 0)で行い，フィードバック(Feed Back……自分自身に対して変調をかける度合)の値も一緒に設定します．このフィードバックはオペレータ1に働くもので，先ほども述べたように歪んだ感じを出してボリュームのある音にするために，ここでは最大値の7にしておきます．

```
A%(0, 0)=7*8+5 ⏎
```

これでフィードバックとアルゴリズムの設定ができました．

次に，音を出すためにAR(Attack Rate)とRR(Release Rate)を設定します．ARは音の立ち上がる速さを設定するので，設定しないと音が出ません．またRRは，Key-offしてからEG出力レベルが減衰していく速さを設定します．このRRを設定しないと，かなり長い間音が続きます．

とりあえず両方とも最大値にしておきます．

```
FOR OP=1 TO 4:A%(OP, 0)=31:NEXT OP ⏎
FOR OP=1 TO 4:A%(OP, 3)=15:NEXT OP ⏎
```

音を出すために，CMD VOICE文でデータを音色として設定しましょう．

```
CMD VOICE A% ⏎
CMD PLAY "V15O4CDEFG" ⏎
```

どうですか．かなり激しい雑音がしたと思います．これはモジュレータのOutput Levelが高すぎるために起こります．モジュレータのOutput Levelが高すぎると，そのモジュレータがキャリアに対してかける変調度が大きくなりすぎて雑音のようになります．

モジュレータのOutput Levelは，作ろうとする音質にもよりますが，最大でも10くらいにとどめておいた方がよいでしょう．

ここでは，モジュレータはオペレータ1だけで，オペレータ1にはフィードバックを最大にかけているので，Output Levelを**30**に設定しておきます．

```
A%(1, 5)=30 ⏎
```

もう1度音を出してみましょう．次のように入力してください．

```
CMD VOICE A% ⏎
CMD PLAY "V15CDEFG"
```

いくらかよい音になったと思います．まだ実際のエレクトリックオルガンの音とは全く違っていますが，細かい修正はあとにして，先に次のパラメータを設定することにします．

次はDR(Decay Rate)を設定します．Key-onされて，ARの値によりEG出力レベルが最高になってからこのレベルが下がっていく速さをDRとSR(Sustain Rate)で決めます．このDRとSRはSL(Sustain Level)によって分かれます．EG出力レベルが最高からSLに達するまでの速さがDRで，SLから出力が0あるいはKey-offされるまでがSRの値になります．このSLを0(減衰量を0とします．したがって，出力レベルは最高値)にすると，EG出力レベルが最高に達した時点からSRになりますので，DRの値は意味を持たなくなり，また，それとは逆にSLを15にすると，EG出力レベルが0になるまでDRの値により減衰していくので，SRの値は意味を持たなくなります．

各オペレータのSLを**15**にします．

```
FOR OP=1 TO 4: A%(OP, 4)=15: NEXT OP ⏎
```

これで大体の準備ができました．それではこれからオペレータを分けて部分的に修正して，音を作り上げていきましょう．

まず，オペレータ4だけONにします．

A%(0, 1)=8 ↵

現在，Multiple(周波数比)の値が0に設定されています．これを1にしておきます．このMultipleは，入力された周波数を何倍にするかを決めるものです．

A%(4, 7)=1 ↵

これで音を出してみましょう．

CMD VOICE A% ↵
CMD PLAY "CDEFG" ↵

あまり音に表情が感じられませんね．これは正弦波によって音が作られているからです．

そこで，次のキャリアであるオペレータ3をONにして，ハモった感じを出すためにMultipleを2にして音を出してみましょう．

A%(0, 1)=4 : A%(3, 7)=2 ↵
CMD VOICE A% ↵
CMD PLAY "CDEFG" ↵

音程は高くなりましたが，やはりこれも正弦波1つなので単純な音ですね．今度はオペレータ3と4をONにして音を出します．

A%(0, 1)=12 ↵
CMD VOICE A% ↵
CMD PLAY "CDEFG" ↵

今度は，原音のオペレータ4と倍音のオペレータ3の音がミックスされてオルガンらしい音色に近づいてきました．次に最後のキャリアであるオペレータ2で高い周波数を出すために，Multipleを10に設定します．

このとき，オペレータ2をONにするのを忘れてはいけません．

A%(0, 1)=14 : A%(2, 7)=10 ↵
CMD VOICE A% ↵
CMD PLAY "CDEFG" ↵

高音部は出るようになりましたが，音程が狂っているように聞こえるので，もう少し下げます．

A%(2, 7)=8 ↵
CMD VOICE A% ↵
CMD PLAY "CDEFG" ↵

これでよくなりましたね．次にモジュレータであるオペレータ１をONにして音を出してみます．

```
A%(0, 1)=15 ↵
CMD VOICE A% ↵
CMD PLAY "CDEFG" ↵
```

多少いびつな音に聞こえますので，オペレータ１のMultipleを **5** にしてOutput Levelを少し下げます．

```
A%(1, 7)=5 : A%(1, 5)=40 ↵
CMD VOICE A% ↵
CMD PLAY "CDEFG" ↵
```

ずいぶんとオルガンらしい音になってきました．しかし，まだ歪んだ感じがしますので，オペレータ２，３のOutput Levelを下げます．

```
A%(2, 5)=20 : A%(3, 5)=30 ↵
CMD PLAY A% ↵
CMD PLAY "CDEFG" ↵
```

まだ歪んだように聞こえますね．これはオペレータ１による変調がオペレータ２，３，４全部に同じようにかかっているためです．そこで，オペレータ１の出力レベルを，音が立ち上がった状態から少し下げるようにします．それには，SLを **2** に設定すればよいでしょう．そして，DRが０ではEG出力レベルが下がらないので，立ち下がりを少し速めの**20**に設定します．

```
A%(1, 4)=2 : A%(1, 1)=20 ↵
CMD VOICE A%
CMD PLAY "CDEFG"
```

これで少し丸みが出てきて，エレクトリックオルガンの感じになってきました．あとは，もう少し音の幅を持たせるためにディチューン(Detune)[注1.]を設定して，周波数のピッチをずらします．

```
A%(1, 8)=3 : A%(2, 8)=-1 ↵
A%(3, 8)=-3 : A%(4, 8)=3 ↵
```

そのあとキーボードレイトスケーリングデプス（Keyboard Rate Scaling Depth）[注2.]を設定して，音程が高くなるほど各Rateによる変化速度が速くなるようにします．

注意

1. ディチューン(Detune)

FM音源のオペレータのピッチを微妙にずらすことによって，音色にうねりや深みを与えることができます．オルガン，エレクトリックピアノ，ストリングスにこの機能を与えることによって，より自然な楽器の感じが得られます．

2. キーボードレイトスケーリングデプス(Keyboard Rate Scaling Depth)

音程によってエンベロープの各Rateを変化させます．自然の楽器のエンベロープはピアノの低音部と高音部を聞いてもわかるように音程によって微妙に異なっています．

このキーボードレイトスケーリングを設定することによって高音部でのRateが速くなり，より自然な楽器の音に近い感じが得られます．

オペレータ2，3，4の各Rate値が最大か，あるいは0で設定されていなければKeyboard Rate Scaling Depthを設定しても意味がないので，ここではオペレータ1だけ**3**にします．

A%(1, 6)=3 ⏎

そして最後に，LFOのパラメータを設定してビブラート(音程を細かく変化させる)やトレモロ(音量を細かく変化させる)などの効果をつけます．しかし，このLFOをあまり深くかけすぎると音程が狂うことがあるので注意してください．

このエレクトリックオルガンの音には，ビブラートだけでトレモロはつけませんが，トレモロをかけるためのパラメータの設定は，LFOスピードが1500〜2500くらい，Amsを3〜8，Amdを低くするとなめらかな効果が得られます．トレモロをかけるときは，最初にこの程度の値を設定しておいて，あとは音色によって修正していくとよいでしょう．

ここではビブラートをかけるために，ピッチ(Pmd, Pms)を変えます．細かいビブラートをかける場合，LFOスピードは3000〜4000くらいがよいでしょう．

エレクトリックオルガンは，それほど強くビブラートをかける必要はないので，Pmsを**5**，Pmdは**10**に設定します．

A%(0, 4)=3000 ⏎
A%(0, 7)=5 ⏎
A%(0, 5)=10 ⏎

次にWave formですが，ビブラートには三角波が適していますので，2を設定して音を聞いてみましょう．LFOはやや高めで長い音にするとわかりやすくなります．

```
A%(0, 2)=2 ⏎
CMD VOICE A% ⏎
CMD PLAY "L1O6C" ⏎
```

もう少しはっきり表すためにPmsを上げます．しかしこのままではかかりすぎるので，Pmdを少し下げることにします．次のようにしてください．

```
A%(0, 7)=6 : A%(0, 5)=8 ⏎
CMD VOICE A% ⏎
CMD PLAY "L1O6C" ⏎
```

これで，ほどよい効果が得られたと思いますが，もっと長い音を出してLFOの効果を確認しましょう．それには"@W"を使うと便利です．

この"@W"は，指定されたときの状態を指定された長さだけ維持します．わかりやすく説明しますと，"@W"が指定されたときにKey-on（音が出ている）状態であれば，そのままKey-onされていて音が出ていますが，指定されたときにKey-offされていれば，そのままKey-offの状態，つまり音が出ない状態が続きます．

```
CMD PLAY "Y40, 240@W1@W1@W1@W1" ⏎
```

以上でエレクトリックオルガンの音のでき上がりです．

今回あげた例は，全く何もない状態から音色を作り上げたので，けっこう大変な作業でしたが，作ろうとする音に近い音，あるいは特徴の似ているものを既存の音色から見つけ出して，その音をもとにして修正を重ね，音を作るようにすれば，もう少し簡単に音を作ることができます．

その方法は，「リファレンスマニュアル」の"CMD VOICE COPY"のサンプルプログラムとして載っていますから，そちらも参照してください．

最後に，N_{88}-BASICシステムディスクの中には効果音や動物の鳴き声を出すプログラムが入っています．拡張子が"voi"となっているファイルを次のようにロードして，実行してください．

```
RUN "<ファイル名>". voi" ⏎
```

ただし，これらのプログラムはN_{88}-BASIC V2が起動した直後の状態で動作します．

4 サウンドユーティリティプログラム

N_{88}-BASICシステムディスクには，サウンド機能を使ったユーティリティプログラムが2つ用意されています．1つは，コンピュータのキーボードをピアノの鍵盤にみたてて音楽を演奏するスーパーキーボードシステム(sks)プログラムです．もう1つは，音色の編集を行うミュージックエディタキット(mek)プログラムです．それぞれのユーティリティプログラムの使い方を説明します．

1. スーパーキーボードシステム(sks)
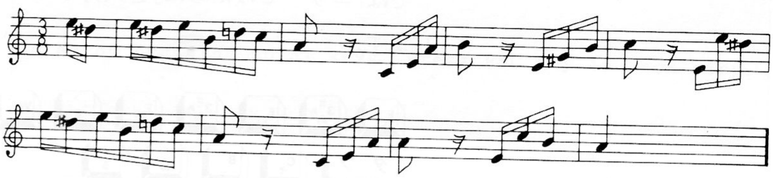
CMD PLAY文を使って音楽を演奏する場合，音の高さ(PITCH)や長さ(RHYTHM)は文字列の形で入力しなければなりません．また，先に音程の文字列を入力しなければ音楽を演奏することができません．これでは音楽をすぐ演奏したいときに不便ですね．

そこで，コンピュータのキーボードをピアノの鍵盤にみたてて，ピアノのように演奏することによってCMD PLAY文の文字列を作ることができたら便利で楽しいと思いませんか？ それを可能にしたのが"sks"プログラムです．

ここでは，ベートーヴェン作の「エリーゼのために」の曲の一部を題材にして，このプログラムの使い方を説明していきましょう．

エリーゼのために

L.V.Beethoven



まず，プログラムを実行させます．V2モードを起動("How many files(0-15)？"が表示されたら，1 ⏎ と入力してください)し，N_{88}-BASICシステムディスクからプログラムをロードした後，実行してください．

注意 sksプログラムは，N_{88}-日本語BASICで実行させることはできません．

しばらくすると，ピアノの鍵盤の図が画面に表れ，次のようなメッセージが表示されます．

```
Input number of sharp (+)/flat(-)?█
```

これは，演奏する曲の調を尋ねています．♯の調ならその♯の数を正の数(1～7)で，♭の調ならその♭の数を負の数(−1～−7)で入力してください．♯，♭両方ともつかない調の場合は 0 [↵] または [↵] を入力してください．

ここで使用する「エリーゼのために」は♯も♭も付かない調なので，[↵] を押します。

すると，次のメッセージが画面に表れます．

```
Press keys to play music. Press ESC key to quit.
```

これで演奏の準備ができました．

画面に表示されている鍵盤の図のとおりにキーを押して演奏してください(ただし，残念ながらこのプログラムでは一度に1音しか音が出せません)．

では「エリーゼのために」を入力してみましょう．次のようにキーを押します．

[E] [3] [E] [3] [E] [.] [W] [/] [,]
[タブ] [C] [B] [,] [.]
[タブ] [B] [K] [.] [/]
[タブ] [B] [E] [3] [E] [3] [E] [.] [W] [/] [,]
[タブ] [C] [B] [,] [.]
[タブ] [B] [/] [.] [,]
[ESC]

＊ [/] と [Q] は同じCの音を発声します．また，テンキーは使うことができません．

す．このデータは後でCMD PLAY文の形でフロッピーディスクにセーブされます．

音を発声するキーの他に，下図に示す特別な働きをするキーがあります．

[タブ]：休符を入力します．
[ESC]：次のステップ（リズムを刻むモード）へ進みます．
[後退]：１つ前の入力を無効にします．
[ ]："！"を表示します．これは音に影響を与えませんので，楽節の区切りを示す目印などに利用できます．
[GRPH]：♮（ナチュラル）にしたい場合，このキーと音階キーを同時に押してください．

以上のキーを使って上手に演奏してください．

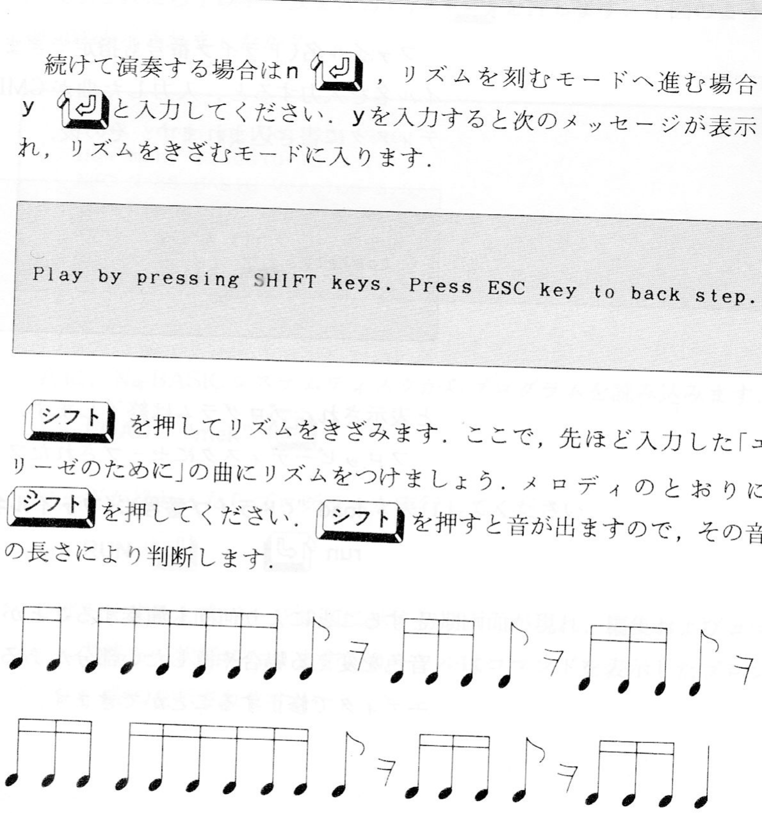
演奏が終わったら[ESC]を押します．すると，次のメッセージが表示されます．

```
Quit(y/n)?█
```

続けて演奏する場合はn [↵]，リズムを刻むモードへ進む場合はy [↵]と入力してください．yを入力すると次のメッセージが表示され，リズムをきざむモードに入ります．

```
Play by pressing SHIFT keys. Press ESC key to back step.
```

[シフト]を押してリズムをきざみます．ここで，先ほど入力した「エリーゼのために」の曲にリズムをつけましょう．メロディのとおりに[シフト]を押してください．[シフト]を押すと音が出ますので，その音の長さにより判断します．



キーを入力すると，画面の左横に，入力した音のデータが表示されます．

［シフト］を押し間違えた場合は［ESC］を押してください．1つ前に戻ります．リズムをきざみ終わると次のメッセージが表示されます．

```
Try again(n/y)? ■
```

リズムをきざみ間違えた場合は y ［↵］と入力して，もう1度最初からリズムをきざみなおしてください．正しくリズムが入力できた場合は n ［↵］を押します．すると，入力した曲が自動的に演奏されます。演奏が終わると次のように表示されます．

```
File name to save? ■
```

ファイル名(ドライブ番号も指定できます)を入力してください．ファイル名を入力すると，入力した曲がCMD PLAY文の形でフロッピーディスクに書き込まれます．その後，

```
coplete.
```

と表示され，プログラムは終了します．

フロッピーディスクにセーブされたファイルは

load "〈ドライブ番号〉:〈ファイル名〉"［↵］

run［↵］

とすることにより何回も演奏することができます．また，入力した曲の音色を変える場合や直したい部分がある場合は，BASICのスクリーンエディタで修正することができます．

注意
1. このプログラムは，入力したリズムどおりに再演奏されない場合があります．
2. B♯を指定した場合，1オクターブ上のCの音になります．また，C♭を指定した場合は，1オクターブ下のBの音ではなく同じオクターブのBの音になります．
3. sksプログラムを実行中("Complete."が表示される前)，[停止]または[コントロール]+[C]を押してプログラムをストップさせたあと，CMD UNLINK文を実行しないでください．

2. ミュージックエディタキット(mek)

"mek"プログラムでは，音色を編集しながら簡単に編集中の音を聞くことができ，また各オペレータの状態を一目で見ることができます．

プログラムの実行は次のように行います．まずN_{88}-BASICシステムディスクをドライブ1に入れてV2モードを起動し，

```
Disk version [Oct 20,1986]
How many files(0-15)?
```

が表示されたら1以上の数を入力し，[↵]を押します．下図のように画面に表示されましたか？

```
Disk version [Oct 20,1986]
How many files(0-15)? 1
NEC N-88 BASIC Version 2.3
Copyright (C) 1981 by Microsoft
***** Bytes free
Ok
```

次に，N_{88}-BASICシステムディスクからプログラムを読み込みます．

LOAD "mek" [↵]

ロードが終わったらプログラムを実行してください．

RUN [↵]

しばらくすると次ページの図に示す初期画面が現れ，編集およびコマンド入力待ちの状態になります．カーソルはコマンドを表示したプロンプト文の横に表示されます．

```
Tone.No  0    Algorithm   2   Feed Back  7    Operator Mask 15
                                op#1    op#2    op#3    op#4
--- RATE ---
AR        [0 - 31]               31      31      31      31
DR        [0 - 31]               12       2      12       5
SR        [0 - 31]                4       4       4       7
RR        [0 - 15]               10       6       6       7
KS          [0 - 3]               0       3       0       2
--- LEVEL ---
ML        [0 - 15]               12      15       1       3
DT        [-3 - 3]                0       1       0      -1
--- OSC ---
SL        [0 - 15]                1      15       0       2
OL       [0 - 127]               32      57      30       0
--- LFO ---
AMS       [0 - 15]                0       0       0       0
WF          [0 - 3]               0
SYNC        [0 - 1]               0
SPEED [0 - 16383]                 0
PMD  [-127 - 127]                 0
AMD  [-127 - 127]                 0
PMS       [0 - 15]                0

<P>lay,<T>one change,<L>oad,<S>ave,<Q>uit ?
```

音色の編集を行う場合は，カーソル移動キー（←→↑↓）や タブ を使ってカーソルを編集したい行へもっていき，数値を指定してから ⏎ を押します．

各音色パラメータにおいて設定できる数値の範囲は次のようになります．

音色のパラメータ	設定できる範囲
Algorithm	0～7
Feed Back	0～7
Operater Mask	0～15
AR(Attack Rate)	0～31
DR(Decay Rate)	0～31
SR(Sustain Rate)	0～31
RR(Release Rate)	0～15
KS(Keyboard Rate Scaling Depth)	0～3
ML(Multiple)	0～15
DT(Detune)	−3～3
SL(Sustain Level)	0～15
OL(Output Level)	0～127
AMS(Amplitude Modulation Sensitivity)	0～15
WF(Wave Form)	0～3
SYNC	0～1
SPEED	0～16383
PMD(Pitch Modulation Depth)	−127～127
AMD(Amplitude Modulation Depth)	−127～127
PMS(Pitch Modulation Sensitivity)	0～15

各音色パラメータの説明については，「リファレンスマニュアル」のCMD VOICE文を参照してください．

入力は1行ずつ行い，各行の入力が終わったら ↵ を押します．ただし，"Tone No."を設定することはできません．"Tone No."を設定する場合はTone Changeモードを指定してください．

ここで編集した音色は，このプログラムの中であらかじめ用意されているTEMP%という配列に割り当てられます．そして，編集した音色は，Playモードに入ることによって聞くことができます．

注意　編集中に【後退】を使用した場合は，ずれて表示されますので，押さないようにしてください．

編集およびコマンド入力待ち状態から違ったモードに入る場合は，"P"(Play)，"T"(Tone change)，"L"(Load)，"S"(Save)，"Q"(Quit)のいずれかの文字を入力します．入力は小文字でも可能です．誤って入力した場合は，【後退】を使って間違えた文字を消去し，それから正しい文字を入力してください．

では，各コマンドについて説明していきましょう．

(1)　Play：音を出す

"P"または"p"をタイプし， ↵ を入力すると音を出すモードになり，現在設定されている音色で音を出すことができます．次のようなプロンプト文が表示されますので，メロディを入力してください．

```
Melody █
```

たとえば，"ドレミ"と音を出した場合は

Melody cde ↵

と入力します．

また，以前にPlayモードでメロディが設定されていた場合，プロンプト文の後に ↵ だけを入力すると，前に設定されたメロディで音を出します．

このPlayモードでは，TEMP%に割り当てられた音色データをチャネル1にセットしますので，単音でのみ音を出すことができます．

演奏が終わると，編集およびコマンド入力待ち状態に戻ります．

(2) Tone Change：音色番号を変える

"T"または"t"をタイプし，↵を入力することによって音色番号を変えることができます。

指定できる数値の範囲は0から61までです。次のようなプロンプト文が表示されますので，音色を編集するときにベースとしたい音の音色番号を入力してください。

音色番号についての説明は，「リファレンスマニュアル」のCMD PLAY文を参照してください．

設定が終わると，設定した音色番号のデータが表示され，再び編集およびコマンド入力待ち状態に戻ります．

```
Tone No. ? █
```

(3) Load：音色データをロードする

"L"または"l"をタイプし，↵を入力すると，以前作成した音色データがTEMP%に割り当てられます．以下のようなプロンプト文が表示されますので，ファイル名(ドライブ番号を指定することもできます)を入力してください．

```
Input File Name█
```

ただし，ここで取り扱えるファイルは，以前"S"(Save)コマンドによって作成されたファイルでなければなりません．指定したファイルのファイル形式が違う場合は"Illegal Program!!"エラーが表示されます．この場合は↵を押すことによって，編集およびコマンド入力待ち状態に戻ることができます．また，指定したファイルがフロッピーディスクに存在しない場合は"DISK ERR!!"エラーが表示されます．この場合も↵を入力することによって，編集およびコマンド入力待ち状態に戻ることができます．以上のようなエラーメッセージが表示された場合は，ファイル名を確認するなどして，もう1度正しいファイル名を入力してください．

音色データの読み込みが終わると，読み込まれたデータが表示され，再び編集およびコマンド入力待ち状態に戻ります．この場合，"Tone No."は"−1"に設定されます．

注意 "S"(Save)コマンドによってセーブされたファイルをBASICのスクリーンエディタで変更したり，あるいはRENUMコマンドを使って行番号を変えた場合，"L"(Load)コマンドでは読み込むことができない場合があります．

(4)　Save：音色データをセーブする

"S"または"s"をタイプし，を入力することによって，音色データをプログラムの形でフロッピーディスクにセーブします．

```
10 DIM TEMP%(4,9)
20 TEMP%(0,0)=58
30 TEMP%(0,1)=15
      ⋮
510 TEMP%(4,9)=0
520 CMD VOICE TEMP%
```

次のようなプロンプト文が表示されますので，ファイル名(ドライブ番号の指定もできます)を入力してください．

```
Input File Name█
```

フロッピーディスクに書き込むときにエラーが起こった場合，"DISK ERR!!"が表示されます．を押すことによって，編集およびコマンド入力待ち状態に戻ります．

"S"(Save)コマンドによってフロッピーディスクに書き込まれたプログラムは，"L"(Load)コマンドによって読み込むことができます．

セーブが終わると，再び編集およびコマンド入力待ち状態に戻ります．

(5)　Quit：プログラムを終了する

"Q"または"q"をタイプしを入力すると，プログラムを終了し，BASICのコマンドレベルに戻ります．

このプログラムで作成した音色はプログラムの形でセーブできますので，CMD PLAY文を使用したプログラムと1本化することにより，自分で作った音色で演奏を楽しむことができます．

5 FM音源の原理/シンセサイザICの構造

これまでは実際に音を出しながらいろいろな機能を体験してきたわけですが，最後にFM音源の原理とシンセサイザICの構造を説明します。しかし，ここまで読み終わった方はもう実際に音楽を演奏したり，自分で音色を作ったりしてきたのですから，この５節は参考資料としてそのまま次の「第13章　入出力装置の管理」に進んでもかまいません．

1．FM音源の原理

FM音源の原理を説明する前に，まずFM音源の基本となるデジタル音源の基礎を説明します．（参考著書「デジタルキーボードハンドブック」日本楽器製造株式会社）

(1) デジタル音源の基礎

デジタル音源は，音を何もない状態から"発生"させるのではなく，コンピュータに使われているのと同じような記憶回路から，いろいろなタイミングで音を読み出し，"再現"する方式をとっています．

この記憶回路にはサイン波だけが，"1"と"0"だけのデジタル符号化されたデータとして入っています．

ここで大切なのは，サイン波がすべてこのような"1"と"0"に置き換わっているということです（図１は説明のための単なる一例で，実際には，サイン波はもっと多くの電圧に分割されていますし，電圧の値もこのようであるとは限りません）．しかし，このような形で記憶回路に入っているのは"サイン波のもと"でしかありません．

というのは，記憶回路にはサイン波を構成する電圧のデータが順番どおりに入っているだけで，その電圧のデータをどういったタイミングで並べるかというデータは入っていないのです．

つまり，ａとｂの間やｂとｃの間など各電圧の間の時間を決定するデータが入っていないので，音として成り立たないのです．もし"時間"が決まらなければ，サイン波の１周期の時間もわからないわけですから，周波数も決まりません．つまり，波形とサイクル（１周期の所要時間）が決まって初めて"音"として成り立つのです．

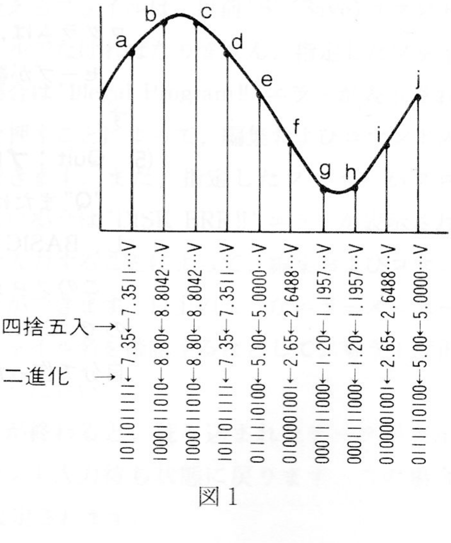


図１

ところで，記憶回路に"時間"のデータを与えるにはどうしたらよいのでしょうか？ ここでちょっと図2を見てください．

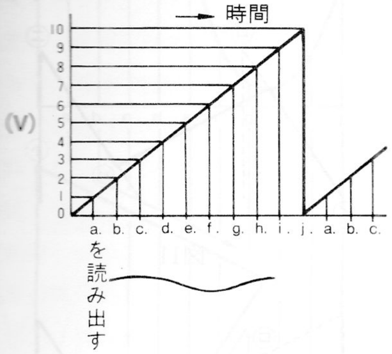


図2

今このノコギリ波を，0Vから上昇し，10Vでストンともとに戻るようなサイクルだと仮定して，これを記憶回路に与えます．そして，ノコギリ波の電圧が1Vのときに記憶回路のａの電圧を出力する，2Vのときにｂの電圧を出力する……というようにして，10Vになったらｊの電圧を出力させるようにしておきます．このような約束にしておくと，ノコギリ波の電圧は直線的に上昇しますから，図3に示すようにａとｂ，ｂとｃといった隣り合った電圧が一定の時間間隔で出力されることになります．

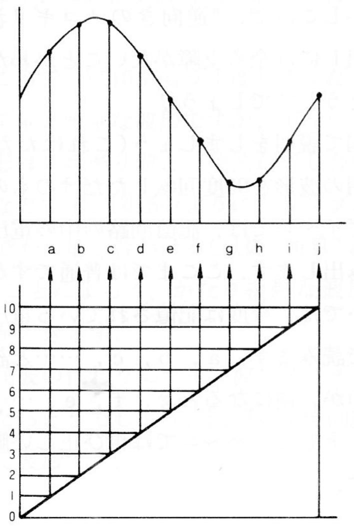


図3

この方法によって，記憶回路からきれいなサイン波が出てきます．そして，この方法で最も重要なのは「読み出されたサイン波の周波数は，読み出しに使ったノコギリ波の周波数と同じである」ということです．

図4を見てわかるように，ノコギリ波の1サイクルとサイン波の1サイクルは常に一致しているので，サイクルが同じであれば周波数も同じになります．

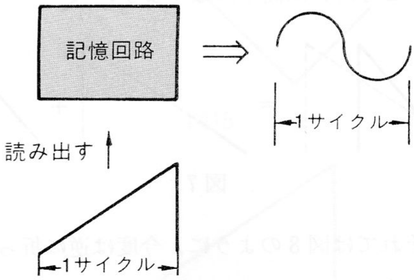


図4

（1サイクルが1ミリセカンドの波形でこの波形が連続している場合，1秒間に同じ波形が1000回繰り返されます．つまり1秒間で1000サイクルですから，これは1000Hzの周波数に当たります）．

(2)　FM音源の原理

FM音源は，デジタル音源のひとつの発展型と考えられます．ですから，"記憶回路からノコギリ波を使ってサイン波を読み出す"のが第一歩となります．

図4をもう1度見てください．ここで，図5のように細工したノコギリ波で記憶回路を読み出してみましょう．出てくるのは図6のようなひしゃげたサイン波になってしまいます．なぜでしょうか？



図5

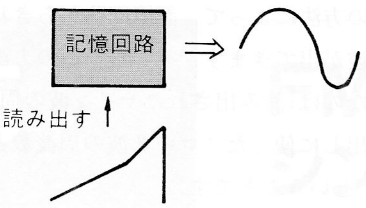


図6

図7を見るとわかるように，折れ曲がったノコギリ波は2つのノコギリ波が合成されたもの，と考えれば簡単になります．

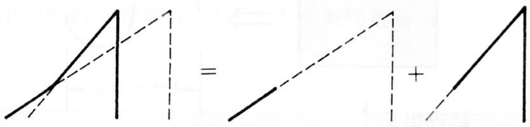


図7

それでは図8のように，今度は逆に折ったらどうなるでしょうか？ さっきとは逆にひしゃげたサイン波になります(図9)．



図8

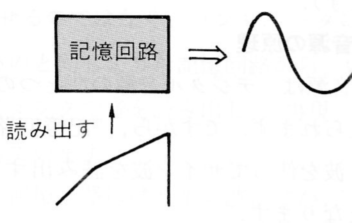


図9

読み出しに使うのが必ずしも正しいノコギリ波でなくてもよいのなら，たとえば図10のようなもっと複雑な波形で読み出してみましょう．この波形で問題なのはロ～ハ間です．

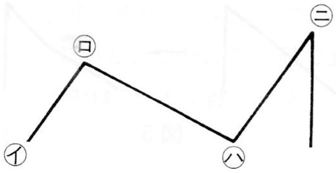


図10

なぜならイ～ロ，ハ～ニ間は，図11のように考えればそれぞれ"ノコギリ波の一部"と見なせますが，ロ～ハでは傾斜が逆です．つまり，逆向きのノコギリ波になってしまいます(図12)．

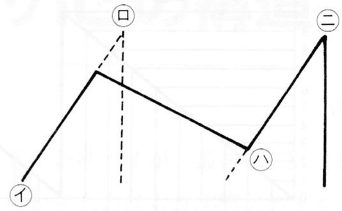


図11

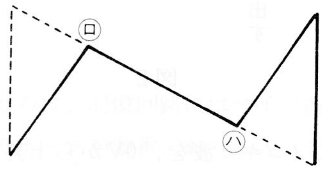


図12

そして，図13が記憶回路から出てくる波形です．これはもうひしゃげすぎていて，とてもサイン波とは呼びがたいですね．

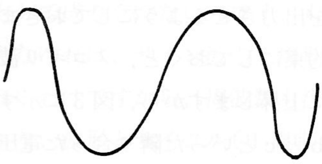


図13

しかしこれで，"逆向きのノコギリ波"でも読み出しには全く支障がないことがわかりました．どうしてでしょう？

図14で説明をしましょう(これはただ，読み出し用の波形を90度回転しただけのものです)．

まずイ～ロでは，記憶回路の中の電圧を順番に読み出します．ここまでは普通ですが，次にロ～ハでは，今度は記憶されている電圧を逆の順番で読みます．a，b，c，……と読んでいたものが，逆になるとg，f，e，……になります．そして，ハ～ニでは再び正しい順番に戻ります．

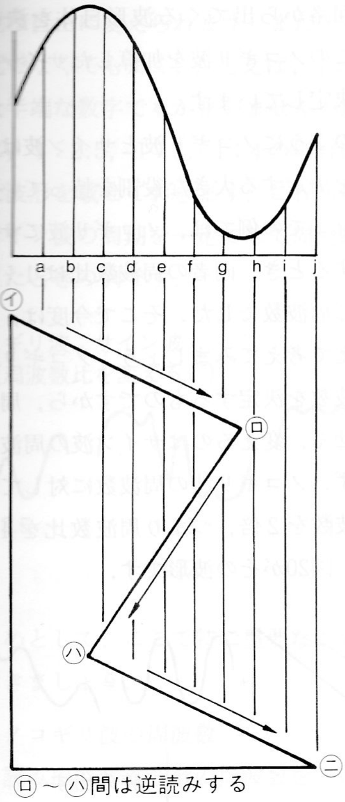


図14

このように記憶回路を読み出すのは，どんな波形でもいっこうにかまいません．そして，読み出しに使う波形によって，記憶回路に入っているサイン波はいろいろな形に変形されて出力されます．これがFM方式の最も大切な基本です．

ところで，図10をもう１度見てください．一見メチャクチャな波形ですが，実をいうと，この波形はノコギリ波と三角波が足し合わされたものなのです．図15のように，シンプルな波形同士でも，足し合わせる(以後"加算"と表現します)ことによって，かなり複雑な波形にすることができます．

三角波の代わりにサイン波をノコギリ波に加算したらどうでしょう．図17が記憶回路から出てくる波形です．

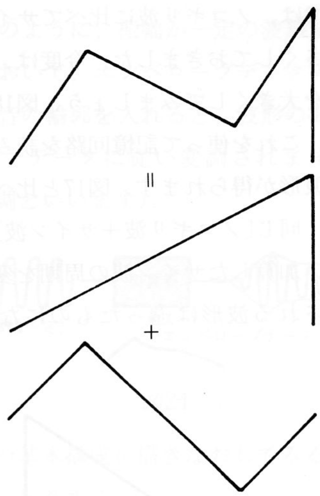


図15

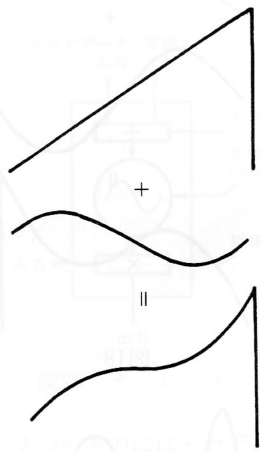


図16

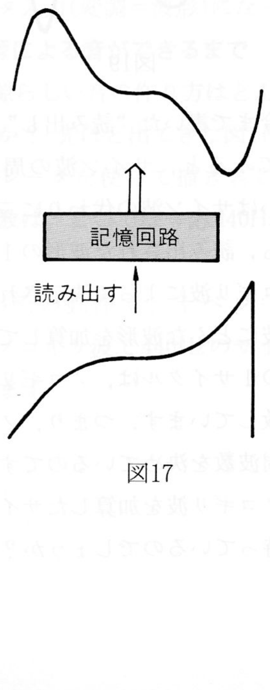


図17

図16では，ノコギリ波に比べてサイン波の周期を小さくしておきました．今度は，サイン波の周期を大きくしてみましょう．図18を見てください．これを使って記憶回路を読み出すと，図19の波形が得られます．図17と比べてみてください．同じ〔ノコギリ波＋サイン波〕で読み出しても，加算したサイン波の周期を変えると，読み出される波形は違ったものになってきます．

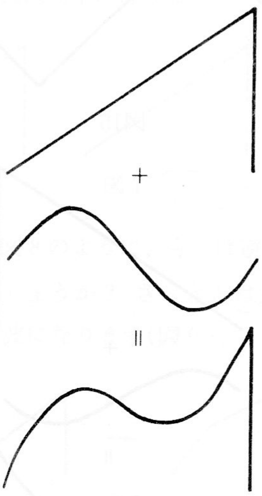


図18

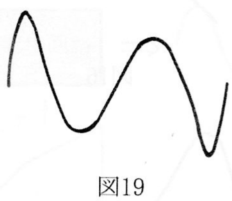


図19

さて，今まで書いた“読み出し”の例をよく見なおしてみると，サイン波の周期を変えたり，あるいはサイン波の代わりに三角波を使った場合でも，読み出された波形の1サイクルはすべてノコギリ波によって決定されています．ノコギリ波にどんな波形を加算しても，出力された波形の1サイクルは，ノコギリ波の1サイクルと一致しています．つまり，ノコギリ波がすべての周波数を決めているのです．

一方，ノコギリ波を加算したサイン波はどんな役割を持っているのでしょうか？　これらは，記憶回路から出てくる波形自体を決めています．このノコギリ波を加算したサイン波が，音色を決定しています．

このようにノコギリ波とサイン波は，それぞれ音を決定する大きな役割を持っています．

これまでの例では，ノコギリ波にサイン波を加算するとき，両者の周波数比は1：1，つまり同じ周波数でした．そこで今度は，周波数比を変えて考えてみましょう．ノコギリ波は出力の周波数を決定するものですから，周波数は変えません．変えるのはサイン波の周波数です．

まず，ノコギリ波の周波数に対してサイン波の周波数を2倍，つまり周波数比を1：2にします．図20がその波形です．

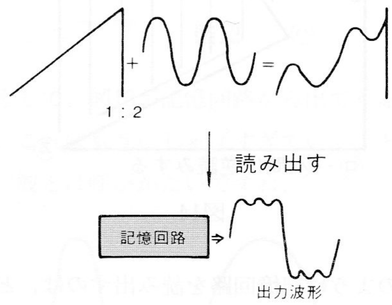


図20

次に，周波数比を1：5にしてみましょう(図21)．1：2のときに比べて，ますます複雑な波形になります．

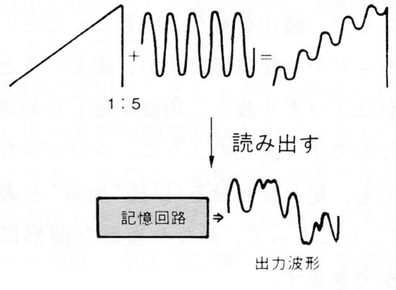


図21

このように，ノコギリ波に同じ周期のサイン波を加算した場合でも，サイン波の周波数を変えていくと，記憶回路から読み出される波形も大きく変化します．ノコギリ波：サイン波の周

波数比は自由に変えられます．また，この比は整数比でなくてもよく，たとえば，1：8.265のような半端な数字でもかまいません．例として図22に，1：2.5，1：3，1：4の場合に読み出された波形を載せておきます．これらは，加算したサイン波の周期を一定にして周波数比だけを変えたときの例です．

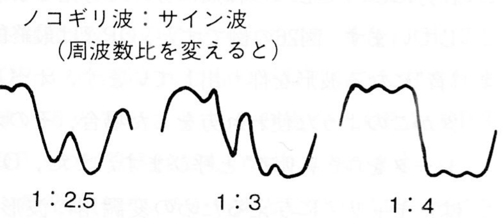


図22

まとめとして，ここで特に重要なことを要約しておきましょう．

(a)　ノコギリ波の周波数
　　読み出される波形の周波数を決定する

(b)　サイン波の周期
　　読み出される波形を決定する

(c)　ノコギリ波とサイン波の周波数比
　　読み出される波形を決定する

これで，FM音源の基本的な知識が身につきました．

(3)　FM音源の基本構成

FM音源の基本構成は，図23のように考えられます．周波数を決めるノコギリ波と，出力の波形を決めるサイン波などが加算器で加算され，それで記憶回路から読み出します．記憶回路からの出力は，一度掛算器を通って出力されます．



図23

図24のように，振幅が一定の波形を掛算器に加えておいて，エンベロープデータ入力に出力される音の輪郭を入れると，波形の振幅はエンベロープデータに従い変調されます（これをFM変調といいます）．

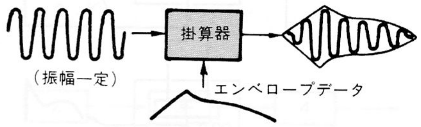


図24

図23の基本構成を描きなおしてみると，図25のようになります．

これを"オペレータ"と呼んでいます．

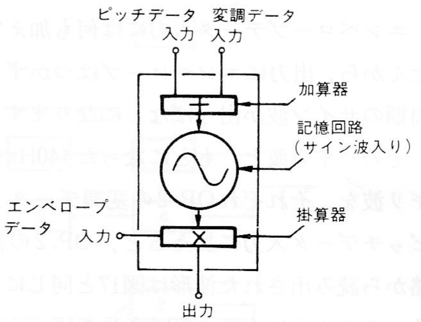


図25　オペレータ

加算器の2つの入力にはそれぞれ名前がついていて，ピッチデータ入力（ピッチ＝周波数）と変調データ入力（変調＝波形）になっています．

(4)　FM音源による音ができるまで

FM音源らしい音の作り方はどんなものなのでしょうか？ 先ほど出てきた図16，17の動作を，オペレータを使って描き表してみましょう．周波数は，ここでは一応440Hzとしておきます．

図26がその動作で，オペレータOP.1は440Hzのノコギリ波で440Hzのサイン波を作り出しています．

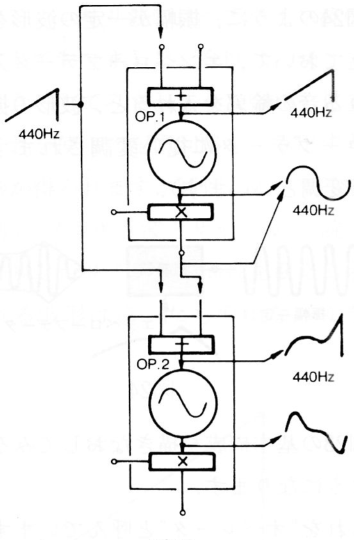


図26

エンベロープデータ入力には何も加えていませんから，出力にエンベロープはつかず，一定周期のサイン波が出っぱなしになります．

そのサイン波と，もとになった440Hzのノコギリ波を，それぞれOP. 2の変調データ入力とピッチデータ入力に加えると，OP. 2の記憶回路から読み出された波形は図17と同じになります．このように実際のFM音源では，ノコギリ波だけを用意しておけば，どんな波形も作り出すことができます．

また図26で，OP. 1のエンベロープデータ入力にある波形の輪郭を入れると，OP. 1の出力のサイン波に輪郭がつき(図27)，これをOP. 2に加えると，最終的に得られる波形(OP. 2の出力)は時間とともに変化し，これは音色の変化に当たります．つまり，音色に変調をかけたことになります．

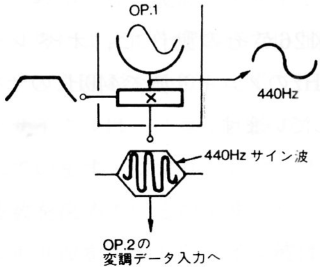


図27

これとは別に，エンベロープデータをOP. 2に加えると，OP.2の出力にも輪郭がつき，これは音量の変化として聴こえます．

FM音源では，このようにオペレータを組み合わせることによって，いろいろな音色を作り出しています．

また，FM音源ではオペレータを，その使われ方によって2つの種類に分けて考えるようにしています．図26の例ですと，OP.2は最終的な"音"になる波形を作り出しています．オペレータがこのような使われ方をした場合，そのオペレータを"キャリア"と呼びます．また，OP. 1は，キャリアに与えるための変調用の波形(この場合はサイン波)を作っています．このようなオペレータを"モジュレータ"と呼びます．

もう1つ，2個のキャリアで音色を作る例をあげておきましょう．図28がそれで，FM音源ではいろいろな周波数のノコギリ波を同時に発生させられることを利用しています．先ほどの図26の例を"直列型"とでも呼ぶなら，こちらは"並列型"になります．

このようなオペレータの接続の仕方を"アルゴリズム"といいます．そして，このアルゴリズムが音色を決定する大きな要素となっています．

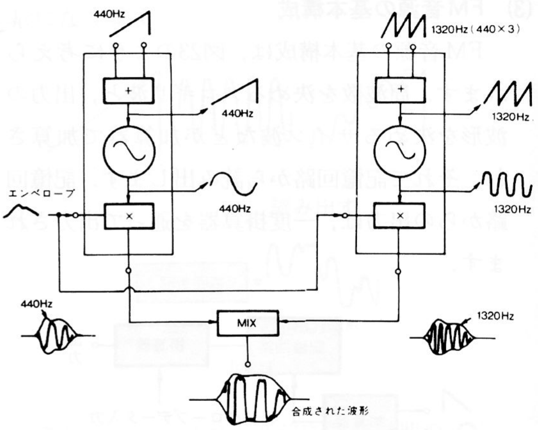


図28

本機に内蔵されているシンセサイザICは，1音に対して4つのオペレータを持ち，図29のように8つのアルゴリズムを選ぶことができます．

図29では，図25のオペレータを箱の形で表し，変調データ入力と出力以外は省略しています．複数の入出力があるものは，それぞれの入出力が加算されていることを意味します．たとえば図29の3のアルゴリズムの場合，OP.2とOP.3の加算されたものがOP.4に入力されます．5のアルゴリズムの場合，OP.2, OP.3, OP.4の加算されたものが出力されます．

周波数はピッチデータとしてオペレータに与えられます．本体に内蔵されているシンセサイザICでは，各々のオペレータに独立したピッチデータを与えていません．基本となる周波数があり，その周波数を何倍かしたものが各オペレータに与えられます．そして，その倍数をMultipleと呼びます．

たとえば440HzのAの音をシンセサイザICに与えたときに，あるオペレータのMultipleが2だった場合，そのオペレータのピッチデータは440Hz×2＝880Hzになります．Multipleの範囲は0〜15までの整数で，0を指定した場合，周波数は$\frac{1}{2}$になります．

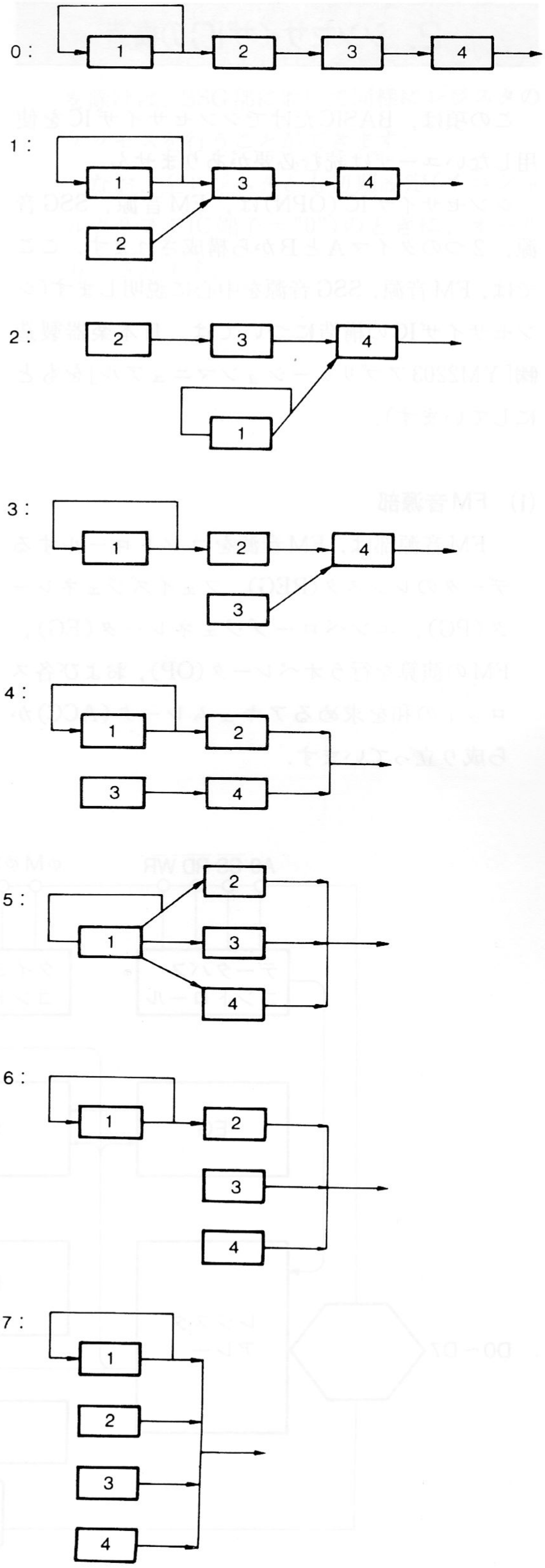


図29

2. シンセサイザICの構造

この項は，BASICだけでシンセサイザICを使用しないユーザは読む必要がありません．

シンセサイザIC（OPN）は，FM音源，SSG音源，2つのタイマAとBから構成されます．ここでは，FM音源，SSG音源を中心に説明します（シンセサイザICの構造については，日本楽器製造㈱「YM2203アプリケーションマニュアル」をもとにしています）．

(1) FM音源部

FM音源部は，FM音源をコントロールするデータのレジスタ（REG），フェイズジェネレータ（PG），エンベロープジェネレータ（EG），FMの演算を行うオペレータ（OP），および各スロットの和を求めるアキュムレータ（ACC）から成り立っています．

(a) レジスタ（REG）

レジスタには，FM音源に対するレジスタと，SSG音源に対するレジスタの2種があります．

FM音源部のレジスタのエリアは，表1のアドレスマップで示されるように，21HからB2Hまでの146バイトの中で割り振られています．ここでいうアドレスとは，OPN内で各レジスタに割り振られたサブアドレスであり，このアドレスのアサインはデータビットD0～D7で行います．したがって，あるレジスタにデータを書き込む場合には，データに先立ってサブアドレスのデータを送り，次にFMデータを送ることになります．サブアドレスについては，いったんあるアドレスを書き込むと，次にアドレスを書き換えるまで，そのアドレスは有効であり，同一のアドレスを何度もアクセスする場合は，アドレスの書き込みは最初に設定するだけで，その後につ

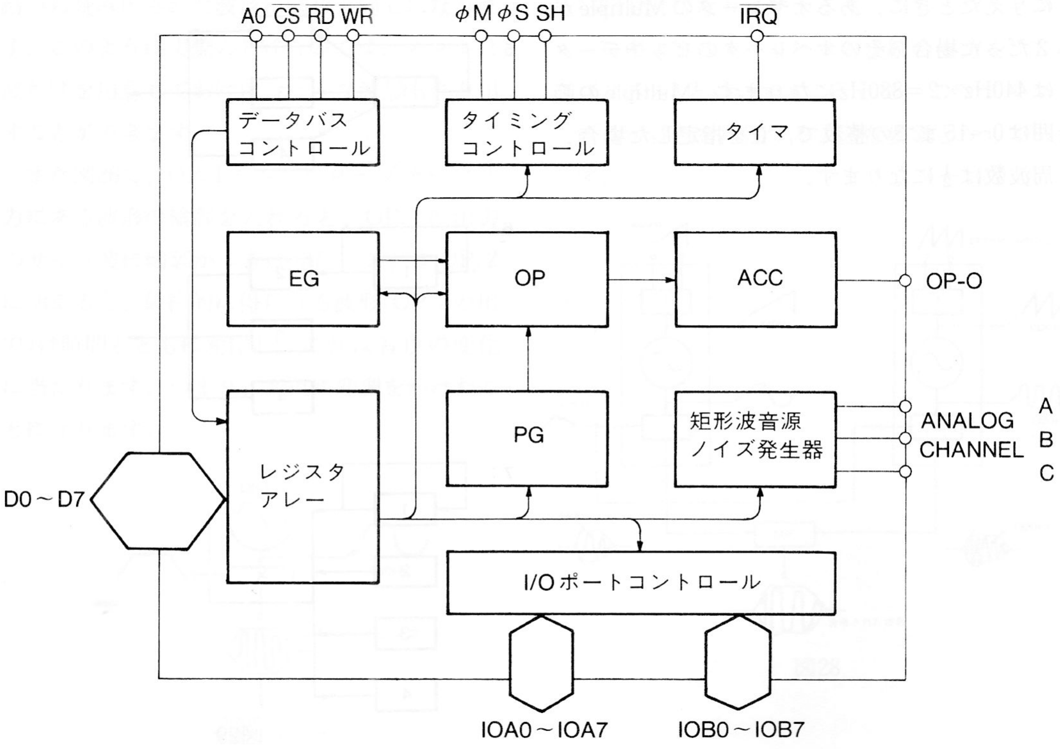


図30　ブロックダイヤグラム

いては不要です．アドレスとFMデータの切り替えはA0入力で行われ，A0＝"0"すなわちLowレベルのときは，データバス上のデータはサブアドレスを意味し，A0＝"1"すなわちHighレベルのときはFMデータとなります．また，FM音源部のデータは書き込みだけしかできないのに対して，SSG部ではレジスタの内容を読むこともできるという点を除けば，SSG部において同様にレジスタのアクセスを行うことができます．

なお，全レジスタとも初期設定(イニシャルクリア：IC端子＝"0")のときに，オール0にされます．

表1　アドレスマップ

※ WRITE DATA ADDRESS	WRITE DATA	COMMENT
21H	TEST	LSIのTEST-DATA
24H	TIMER-A	TIMER-Aの上位8bits
25H	TIMER A	TIMER-Aの下位2bits
26H	TIMER-B	TIMER-BのDATA
27H	MODE / RESET B A / ENABLE B A / LOAD B A	TIMER-A/BのControlおよび3CHのMode
28H	SLOT / CH	Key-on/off
2DH		プリスケーラをSet
2EH		1/3, 1/6分周の選択
2FH		分周器を1/2分周にセット
30H–3EH	DT / MULTI	Detune/Multiple (33H, 37H, 3BHのAddressは無し)
40H–4EH	TL	Total Level (43H, 47H, 4BHのAddressは無し)
50H–5EH	KS / AR	Key Scale / Attack Rate (53H, 57H, 5BHのAddressは無し)
60H–6EH	DR	Decay Rate (63H, 67H, 6BHのAddressは無し)
70H–7EH	SR	Sustain Rate (73H, 77H, 7BHのAddressは無し)
80H–8EH	SL / RR	Sustain Level / Release Rate (83H, 87H, 8BHのAddressは無し)
90H–9EH	SSG-EG	SSG-Type Envelop Control (93H, 97H, 9BHのAddressは無し)
A0H A1H A2H	F-Num. 1	F-Numbers/BLOCK
A4H A5H A6H	BLOCK / F-Num. 2	
A8H A9H AAH	3CH＊F-Num. 1	3CH-3slot F-Numbers/BLOCK
ACH ADH AEH	3CH＊BLOCK / 3CH＊F-Num. 2	
B0H B1H B2H	FB / CONNECT	Self-Feedback/Connection

※READ/WRITE DATA

ADDRESS			COMMENT
00H	Fine Tune		Channel-A Tone Period
01H		Coarse Tune	
02H	Fine Tune		Channel-B Tone Period
03H		Coarse Tune	
04H	Fine Tune		Channel-C Tone Period
05H		Coarse Tune	
06H		Period Control	Noise Period
07H	IN/OUT IOB IOA	/Noise /Tone	/ENABLE
08H		M Level	Channel-A Amplitude
09H		M Level	Channel-B Amplitude
0AH		M Level	Channel-C Amplitude
0BH	Fine Tune		Envelop Period
0CH	Coarse Tune		
0DH		C ATT ALT HLD	Envelop Shape, Cycle
0EH	I/O Port-A		I/O Port Date
0FH	I/O Port-B		

※READ DATA

ADDRESS			COMMENT
00H ~ 0FH	BUSY	FLAG B A	Status

① TEST : ADDRESS [21H]

このアドレスは，OPNをテストするために設けられたのであり，常に"0"レベルにしておかなければなりません．

② TIMER

タイマは2個あり，1つは10ビットプリセッタブルタイマ(TIMER-A)で，もう1つは8ビットプリセッタブルタイマ(TIMER-B)です．各タイマは始動・停止およびフラグの制御が可能です．

● TIMER-A : ADDRESS［24H］・［25H］

タイマAは，アドレス24Hと25Hの10ビットから成り，この10ビットの値をプリセット値として，カウンタがオーバーフローを生じたときにタイマAのフラグを立て，同時にプリセット値をロードします．

タイマAは，通常のタイマ機能以外にCSMのコントロールとしても働きます．この場合は，オーバーフローが生じたときのみチャネル3の各スロットをONにして，チャネル3の全スロットの音を出します．これにより，複合正弦波合成による音声合成が可能になります．

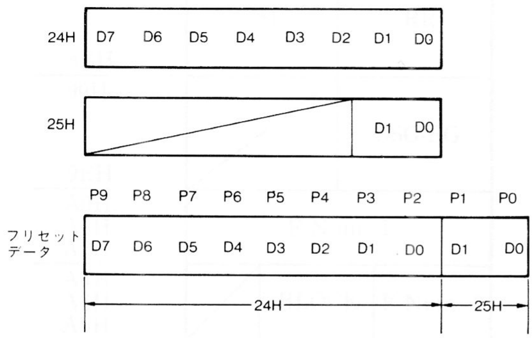


図31

● TIMER-B : ADDRESS［26H］

タイマBは8ビットのプリセッタブルタイマであり，タイマと同様にカウンタのオーバーフロー時にフラグを立てます．

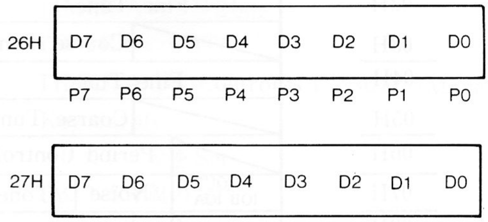


図32

● タイマコントロールおよびチャネル3のMODE設定：ADDRESS［27H］

アドレス27Hでは，タイマA，タイマBの始動・停止・フラグの制御をD0～D5のビットで行います．また，D6, D7の2ビットでチャネル3のモードを決定します．

各ビットの意味

D0：タイマAの始動・停止の制御をします．このビットに"1"が立ったときに，その立ち上がりで，タイマAのプリセット値をロードしてタイマAを動かします，そして，このビットが"0"になるとタイマAは停止します．

D1：タイマBの始動・停止の制御ビットで，働きはタイマBに対してD0と同様なことを行います．

D2：このビットに"1"が立つと，タイマAのオーバーフローを受けてAフラグに"1"を立てます．"0"のときはオーバーフローは受けつけません．

D3：このビットはタイマBに対してD2と同一の働きをします．

D4：このビットに"1"が立つと，Aフラグをリセットします．ただし，このビットの"1"は保持されず，Aフラグをリセットすると，クリアされます．

D5：Bフラグをリセットします．リセットデータは保持されません．

D6, D7：この2ビットは上記のD0～D5とは違った意味を持っており，チャネル3のモードを設定します(表2)．

表2

MODE	D7	D6	内　　容
通　常	0	0	チャネル3は通常の発音モードです．F-Numberは$A2・$A6で指定されます．
効果音	0	1	チャネル3のF-Numberは各スロットごとに指定できます．1スロットは$A9・$AD, 2スロットは$AA, $AE, 3スロットは$A8・$ACのデータを使います．
音声合成	1	0	チャネル3のF-Numberは10の場合と同様ですが，MODEはCSMの音声合成モードとなり，チャネル3のKey-on/offはタイマーAでコントロールされます．

③　Key-on/off：ADDRESS［28H］

このレジスタによって，FM12スロットのKey-on/offがコントロールされます．レジスタの下位2ビットがチャネルを指定します．そして，上位4ビットが各チャネルのスロットを指定できます．指定したチャネルスロットに"1"を立てるとKey-onとなり，"0"を書き込むとKey-offとなります．

スロット割り当て

D4	第1スロットのON/OFF
D5	第2スロットのON/OFF
D6	第3スロットのON/OFF
D7	第4スロットのON/OFF

例

	D7	D6	D5	D4	D3	D2	D1	D0
チャネル1の全スロットをON	1	1	1	1	*	*	0	0
チャネル2の全スロットをOFF	0	0	0	0	*	*	0	1
チャネル3の1,2スロットのみON	0	0	1	1	*	*	1	0

＊はDON'T CARE

図33

④　プリスケーラ機能：ADDRESS［2DH］・［2EH］・［2FH］

ここでは，FM・SSG各音源の周波数をコントロールします．この3つのアドレスに関してはデータビットはなく，アドレスを指定するだけです．分周波とアドレスの関係は表3のようになります．ただし，FM音源の出力はこの周波数の倍で出されます．

このプリスケーラの注意点として，初期設定時は，FM音源に対しては$\frac{1}{6}$の分周数に設定されています．

表3　入力クロックと内部クロックの関係

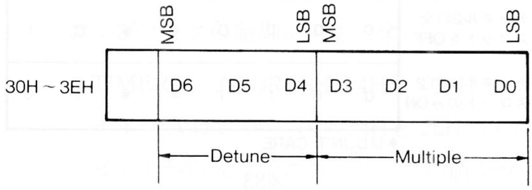
2D	2E	2F	FM音源の分周数	SSG音源の分周数	OPNに入力できる最大周波数
—	—	A	1/2	1/1	1.4MHz
A	—	—	1/6	1/4	4.2MHz
A	A	—	1/3	1/2	2.1MHz

(Aはそのアドレスを入力し，—は入力しないことを意味します．)

⑤ Detune/Multiple : ADDRESS［30H～3EH］

Multiple は F-Number と Detune で得られる位相情報に対して，表4で表されるような倍率の情報であり，Detune は F-Number にわずかの周波数ズレを与える情報です．

表4　倍率



Multiple情報	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
倍　率	1/2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Detune/Multiple より SSG-EG までの7種類の FM 制御情報は，各スロットごとに決めることができます．各アドレスとスロットチャネルの対応は，表5のとおりです．表からもわかるように，チャネルとアドレスの下位2ビットが対応して，アドレスの下位2ビットが"11"になることはありません．また，2スロットと3スロットの順序が逆になっています．

表5　アドレスとチャネル・スロットの関係

ADDRESS	チャネル	スロット	ADDRESS	チャネル	スロット
＊0H	1	1	＊8H	1	2
＊1H	2	1	＊9H	2	2
＊2H	3	1	＊AH	3	2
＊4H	1	3	＊CH	1	4
＊5H	2	3	＊DH	2	4
＊6H	3	3	＊EH	3	4

（＊は3より9までの値）

また，F-Number/Block と Self-Feedback/Connection のデータはチャネルごとに決められるものであり，そのデータは対応する4スロットに共通です．アドレスとチャネルの関係は次のとおりです．

表6　アドレスとチャネルの関係

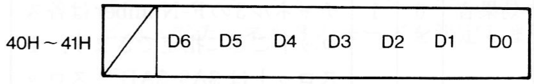
ADDRESS	チャネル
＊0H, ＊4H	1
＊1H, ＊5H	2
＊2H, ＊6H	3

（＊はAあるいはB）

⑥ Total Level : ADDRESS［40H～4EH］

トータルレベルとは，エンベロープジェネレータの出力に対して，減衰量を加算し変調度（音色）および音量の制御をするために用いられます．減衰量は最小分解能を0.75dB として，各ビットの重み付けは表7のとおりです．

表7　トータルレベルでの減衰量



データビット	D6	D5	D4	D3	D2	D1	D0
減衰量（dB）	48	24	12	6	3	1.5	0.75

トータルレベルには，もう1つの働きがあります．それは，CSM のモードを選択した場合です．このときトータルレベルは3チャネルの音（CSM 合成音）に対し，エンベロープのイニシャルレベルとなります．したがって，エンベロープの途中でトータルレベルを変えても，エンベロープの減衰量は変化せず，次の Key-on 時のレベルが変わります．

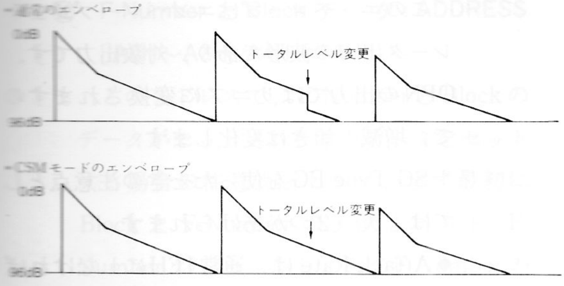


図34

⑦　Key Scale/Attack Rate : ADDRESS [50H～5EH]

Key Scaleは，エンベロープのRateを音程によって変化させるために設けられています．Key Scaling後のRateは，次式によって表されます．式中のRは入力したRateであり，RksはKey Scaling量で，この値は表8のとおりです．

Rate＝2＊R＋Rks

(R＝0のとき，Rate＝0)

Attack Rateは，音の立ち上がりの時間を設定します．

なお，エンベロープ形状は「(2)　SSG音源部」の⑤のエンベロープジュネレータの項を参照してください．

表8　RateのKey Scale

KS \ Key-Code	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
2	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7
3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

KS \ Key-Code	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
1	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7
2	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15
3	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Key-Codeについては(b-1)を参照．

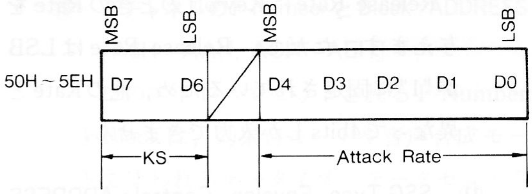


図35

⑧　Decay Rate : ADDRESS [60H～6EH]

Decay時のRateを設定します．

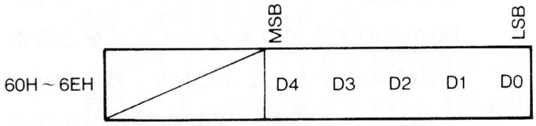


図36

⑨　Sustain Rate : ADDRESS [70H～7EH]

Sustain時のRateを与えます．

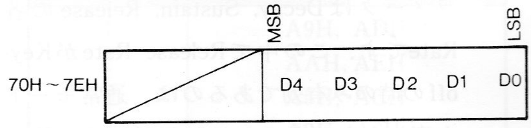


図37

⑩　Sustain Level/Release Rate : ADDRESS [80H～8EH]

Sustain LevelはDecay Rateと，Sustain Rateの切り替え点を与えます．その重み付けは表9で表され，値は減衰量を示します．

表9　サスティンレベルと重み付け

データビット	D7	D6	D5	D4
減衰量(dB)	24	12	6	3

(ただし．D4～D7がすべて"1"の場合は，93dBとなります)．

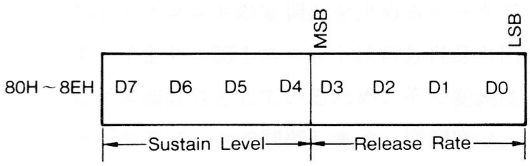


図38

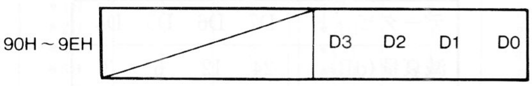
Release RateはKey-offのときのRateを与えます．ただし，Release RateはLSBが"1"に固定されているため，他のRateと異なって4bitsしか入力できません．

⑪ SSG-Type Envelop Control : ADDRESS [90H～9EH]

SSG-Type Envelopとは，後述のSSG音源で作られるエンベロープと同一形状を与える機能です．この4ビットのデータ中，D3がこのタイプのエンベロープをとるか否かのスイッチとなっており，残り3ビットがデコードされ，図40の形状のエンベロープを発生します．

このエンベロープが使われた場合の注意点は，Rateの使い方が通常の場合とは異なっていることです．このモードでは，パラメータはDecay, Sustain, Releaseの各Rateです．この中でRelease RateがKey-offの時のみ有効であるのは，通常モードと変わりありません．したがって，Key-on中に音量が増加したり減少したりする場合の制御は，DecayおよびSustainの2つのRateでなされます(音量の増加は減少の場合を逆にたどっていく形になります).す)．

もう1つ通常モードと異なる点は，減衰時間の違いです．通常モードより変化が速くなります．



＊ Attack Rateは1FHに固定されていなければなりません．

図39

このエンベロープはエンベロープジェネレータ出力の波形であり，対数出力です．OPNの出力ではリニアに変換されますので，増減の傾きは変化します．

SG-Type EGを使ったときの注意点としては，次の2つがあげられます．

- Attack Rateは，通常1FHにしなければなりませんが，08H, 09H, 0BHなどは普通のエンベロープ時のRATEで変化します．その他のものについては十分な制御はできません．
- Decay Rate, Sustain Rate, Sustain Levelは，エンベロープがUP時，またはDOWN時では異なります．

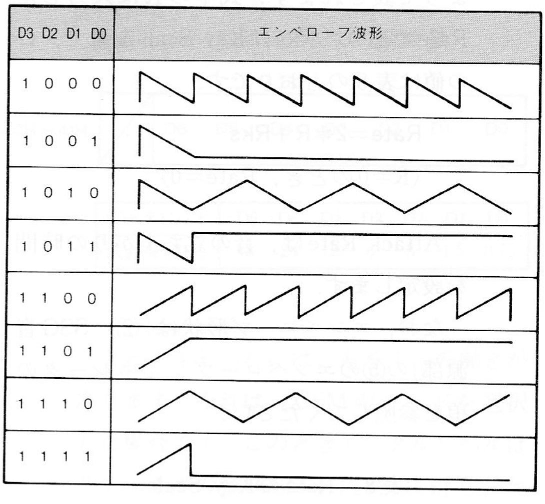


図40

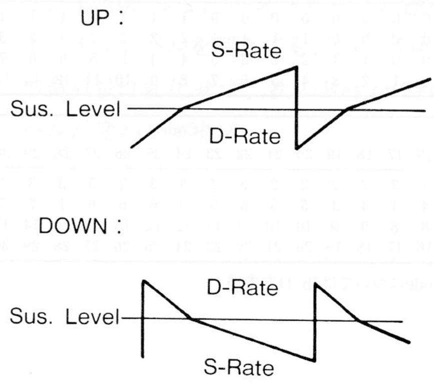
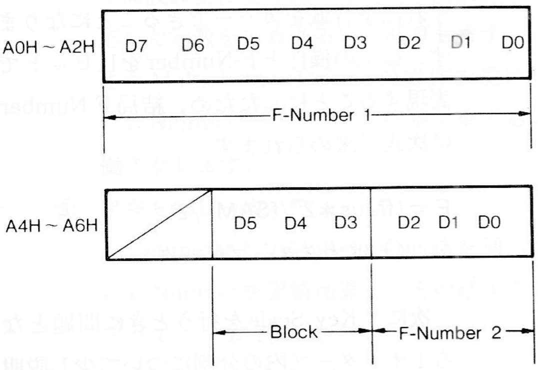


図41

⑫　F-Number と Block データ：ADDRESS ［A0H〜A2H, A4H〜A6H］

このアドレスで，F-Number と Block のデータがセットされます．データをセットする場合の注意点としては，まず最初に Block と F-Number の上位 3 ビット（F-Number 2)をセットし，次にF-Numberの下位 8 ビット(F-Number 1)を入力します．

F-NumberとBlockの内部のレジスタへのローディングは，F-Number 1がセットされたときのみ動作しますから，データの入力の順序が異なりますと，Block と F-Number 2のデータは違ったデータとなります．



例　音程データ(@ϕ M＝1.2MHz)

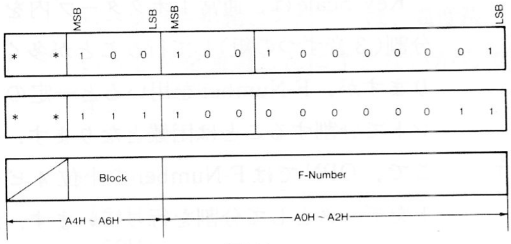


図42

⑬　3チャネルのF-NumberとBlock：ADDRESS ［A8H〜AAH, ACH〜AEH］

このアドレスでセットされるF-NumberとBlockは，効果音モードや音声合成モードで使われるデータです．データセットの方法は⑫で述べた方法と同様で，先に Block, F-Number 2 のデータを，次に F-Number 1のデータをセットします．

ここでセットしたデータとスロットの関係は，次のようになります．

下のスロットとアドレスとの関係は，効果音モードと音声合成モードのときに適用されるのであって，通常モードでは全スロットとともA2HとA6Hのアドレスで与えられます．

スロット番号	F-Number/Blockのアドレス
1	A9H, ADH
2	AAH, AEH
3	A8H, ACH
4	A2H, A6H

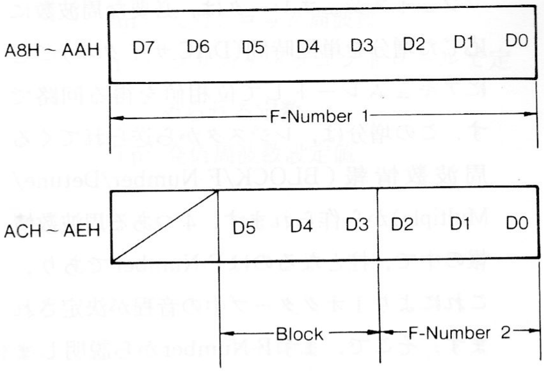


図43

⑭　Self-Feedback/Connection：ADDRESS ［B0H〜B2H］

Self-Feedbackというのは，各チャネルの第1スロットの変調度を決めるデータです．つまり，第1スロットは自分自身の出力を変調信号としているため，その変調度合をこのデータで制御します．変調度は，表10に示すとおりです．

表10 変調度

D3～D5	0	1	2	3
変調度	OFF	π/16	π/8	π/4
D3～D5	4	5	6	7
変調度	π/2	π	2π	4π

Connectionは4つのスロットをどのように変調波と被変調波に分け，どういうふうに組み合わせるかを指定するデータです．それらの組み合わせは全部で8つあり，その構成については「リファレンスマニュアル　第3章　CMD VOICE」のアルゴリズム表を参照してください．

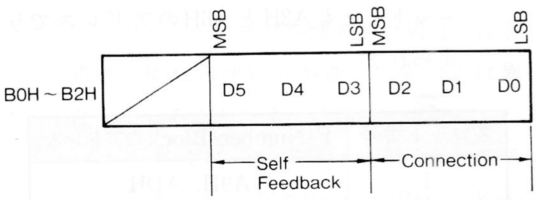


図44

(b) フェイズジェネレータ(PG)

フェイズジェネレータは，必要な周波数に応じた増分を単位時間(DACサイクル)ごとにアキュムレートして位相値を得る回路です．この増分は，レジスタから送られてくる周波数情報(BLOCK/F-Number/Detune/Multiple)から作られます．4つある周波数情報の中で，柱となるのはF-Numberであり，これにより1オクターブ中の音程が決定されます．そこで，まずF-Numberから説明します．

① F-Number/BLOCK

位相の増分は，発音したい周波数と，入力クロック周波数(サンプリング時間)が決まると，すぐに求めることができます．これは次式になります．

$\triangle P = fMus * 2^{20} / fSAM$

$fSAM = fM / (12 * n)$ n=2, 3, 6

fMus ：発音したい周波数

fSAM：サンプリング周波数

fM ：入力クロック周波数

n ：プリスケーラコントロールで得られる定数[217ページ「④プリスケーラ機能」参照]

これで位相の増分は求まりますが，この増分を管理するのはビット数が多くて大変です．そこで，増分は1オクターブ分のデータのみを管理できるビット数に制限し，指定したオクターブごとにその増分をシフトすることにより，位相を求めることとします．

これにより，オクターブ情報の指定をBlockデータに，増分をF-Numberに入力すれば全音域をカバーできることになります．以上の操作とF-Numberを11ビットで表現することにしたため，結局F-Numberは次式で求められます．

$F = (fMus * 2^{20} / fSAM) / 2^{b-1}$

(bはオクターブデータ)

次に，Key Scaleを行うときに問題となる1オクターブ内の分割について少し説明します．

Key Scaleは，通常1オクターブ内を4分割(3音ずつの組)して行うことが多くありますが，F-Numberを用いると一定のセントで分割することは困難となります．そこで，OPNではF-Numberの上位4ビットをデータとして分割を行っています．ただし，この方法では入力周波数によって分割が異なります．

1オクターブ内の周波数分割

N4＝F11

N3＝F11・(F10＋F9＋F8)＋F11・F10・F9・F8

以上の操作で得たN4, N3の2ビットと，オクターブデータの3ビットの計5ビットでKey Scaleを行います．

	Block	Note
	B3 B2 B1	N4 N3
Block, Noteを分離した重み付け	D2 D1 D0	D1 D0
Block, Noteをまとめた場合の重み付け	D4 D3 D2	D1 D0

図45

② Detune/Multiple

レジスタの項でも説明したように，Detuneは与えられたF-Numberに対して周波数変位を与えるために，周波数に応じた定数を加算あるいは減算します．またMultipleは，Detuneの操作を受けたF-Numberに指定された倍率を乗ずる働きをします．

③ アキュムレータ

DetuneおよびMultipleの回路を通ったF-Numberを累積加算し，その結果をオペレータに送ります．このアキュムレータは加算のみを行い，オーバーフローが生じていても無視します．オーバーフローが起こったことは，弧度法でいう2πを通過したにすぎないからです．

(2) SSG音源部

SSG音源部はDACを内蔵した音源であり，そのレジスタアレーにデータを書き込むだけで楽音を発生します．SSG内部のレジスターは，FM音源部と異なり，READ/WRITEが可能です．以下，各レジスタごとにその機能を説明していきます．

① トーンジェネレータコントロール：ADDRESS［00H～05H］

SSG部も，FM音源部と同様に3チャネルの楽音を発生することができます．この3チャネルの楽音の周波数を設定するのがこのレジスタです．楽音周波数は12ビットで構成されるため，1チャネルにつき2つのアドレスを必要とし，そのビットの対応は下記のとおりです．また，発信周波数は次式で求められます．なお，出力波形は1:1の短形波です．

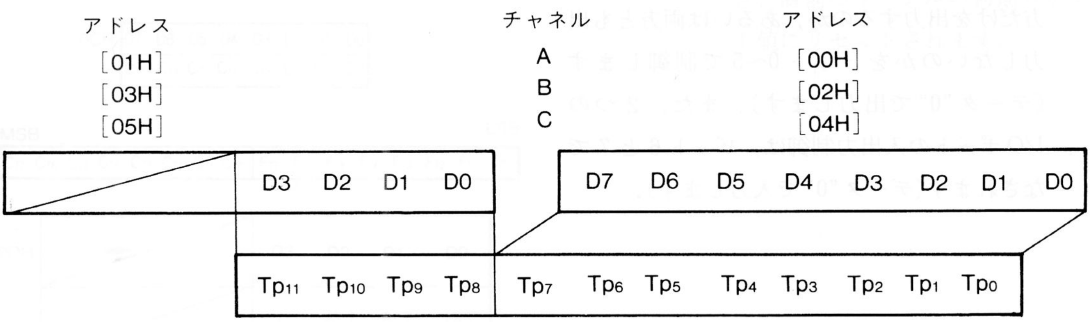
ftone=fm/(8＊n＊Tp)

n=1, 2, 4

fm：入力クロック周波数

n ：プリスケーラコントロールで定められる定数

Tp：発信周波数設定値



この分周においてTp=0とTp=1は同一値となります．

図46

② ノイズジェネレータコントロール：ADDRESS［06H］

ノイズ源は系列長の疑似ランダムノイズです．そして，この系列の発生をコントロールするのがこのレジスタの働きです．系列発生周期（ノイズ周波数）はレジスタの下位5ビットで決められ，次式で求めることができます．

$fN = fM/(8*n*Np)$

$n = 1, 2, 4$

fM：入力クロック周波数
n ：プリスケーラコントロール設置値
Tp：ノイズ周波数設定値

この分周においてNp=0とNp=1は同一値となります．

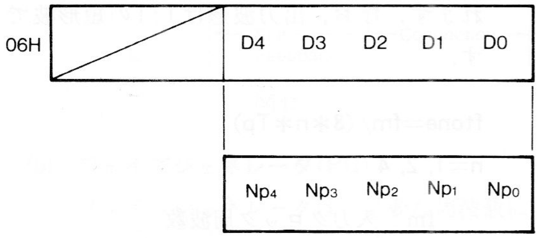


図47

③ ミキサI/Oコントロール：ADDRESS［07H］

このレジスタは3つのノイズ／トーンミキサと，2つの汎用I/Oポートのコントロールをします．ミキサは3チャネルごとに，ノイズとトーンともに出力するのか，いずれか一方だけを出力するのか，あるいは両方とも出力しないのかをビット0～5で制御します（データ"0"で出力します）．また，2つのI/Oポートの入出力制御は，ビット6と7でなされます（データ"0"で入力します）．

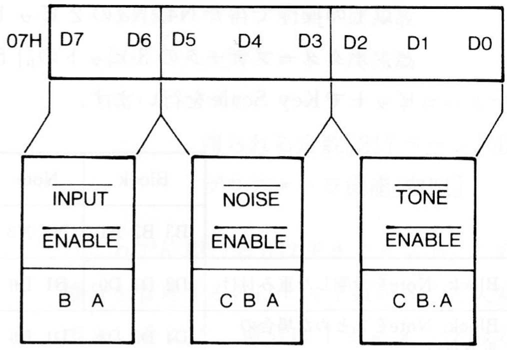


図48

④ 振幅コントロール：ADDRESS［08H～0AH］

A, B, Cの3つのチャネルに対応するD/Aコンバータの振幅は，このレジスタによって決められます．レジスタのビット4が振幅のモードを決めるデータとなっており，"0"のとき固定振幅モードとなり，残りの4ビットで決められる値をとります．また，"1"の場合は可変振幅モードとなり，後述するエンベロープコントロールの出力で決まる振幅となります．なお，D/Aコンバータは，入力5ビットの能力を持っており，固定振幅モード時の4ビットの振幅データは，D/Aコンバータの入力上位4ビットに対応します．

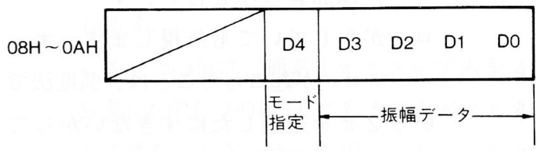


図49

⑤ エンベロープジェネレータコントロール：ADDRESS［0BH～0DH］

変化のあるエンベロープを発生するために，2つのエンベロープコントロールがあります．その1つは，アドレス0BHと0CHで定められるエンベロープ周波数を制御することであり，もう1つは，アドレス0DHのエンベロープの形状を制御することです．

まず，エンベロープ周波数について説明します．指定されたアドレスの2バイト16ビットのデータで入力周波数を分周することにより，エンベロープ周波数は求まります．それは次式で与えられます．

$$fE=fM/(128*n*Ep)$$
$$n=1,\ 2,\ 4$$
$$Ep=256*CT+FT$$

fM ： 入力クロック周波数
n ： プリスケーラコントロール設定値
CT ： 0CHで与えられるエンベロープ周期設定値
FT ： 0BHで与えられるエンベロープ周期設定値

このエンベロープを作るカウンタの上位5ビットがD/Aコンバータへ送られ，振幅データとなります．

0BH	D7	D6	D5	D4	D3	D2	D1	D0
	F_{T7}	F_{T6}	F_{T5}	F_{T4}	F_{T3}	F_{T2}	F_{T1}	F_{T0}

0CH	D7	D6	D5	D4	D3	D2	D1	D0
	C_{T7}	C_{T6}	C_{T5}	C_{T4}	C_{T3}	C_{T2}	C_{T1}	C_{T0}

MSB　　LSB

C_{T7}	C_{T6}	C_{T5}	C_{T4}	C_{T3}	C_{T2}	C_{T1}	C_{T0}	F_{T7}	F_{T6}	F_{T5}	F_{T4}	F_{T3}	F_{T2}	F_{T1}	F_{T0}

0DH		D3	D2	D1	D0
		C	ATT	ALT	HLD

図50

次にエンベロープの形状について説明します．先に述べたように，エンベロープはエンベロープカウンタの上位5ビットで決まります．したがって，ここでコントロールするのは，この上位5ビットのカウントの方法ということになります．アドレス0DHの下位4ビットがこれを決定します．各ビットの機能は次のように定義され，それらの組み合わせにより，表11で与えられるエンベロープの形状が得られます．

C ： "1"にセットされたときはHLDのデータで定められる状態になり，"0"の場合は，エンベロープジェネレータは1サイクル経過後すべて"0"にリセットされます．

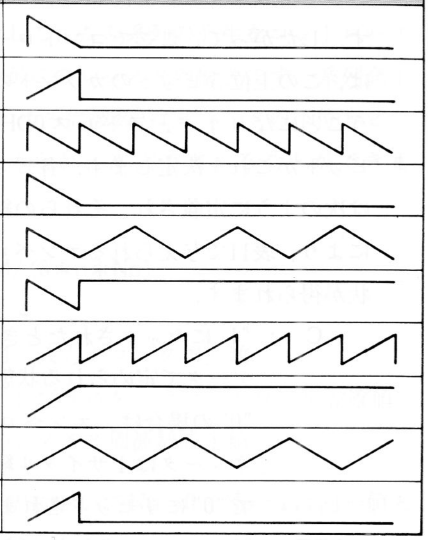
ATT ： "1"にセットされたときは，エンベロープジェネレータはすべて"0"からすべて"1"へカウントアップし，"0"であればすべて"1"からすべて"0"へカウントダウンします．

ALT ： "1"にセットされたときは，各サイクルごとにエンベロープカウントの方向を逆転します．

HOSD： "1"のときは，エンベロープカウンタを1サイクルカウントした後，カウンタの最終値をホールドします．ただし，ALTも"1"のとき，直前のサイクルの初期カウント値にリセットされます．

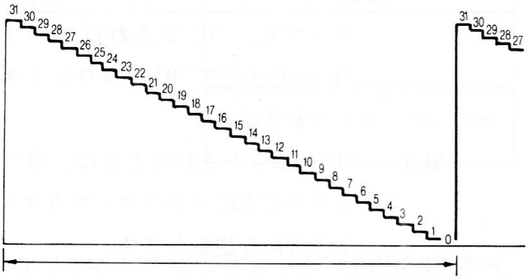
表11　SSG部のエンベロープ波形

D3 D2 D1 D0 C ATT ALT HLD	エンベロープ形状
0 0 X X	
0 1 X X	
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	



X：Don't care

表12　1000データの詳細



第13章

入出力装置の管理
(ファイルディスクリプタ)

BASICでは，入出力装置との情報のやりとりを「ファイル(書類)」という概念で行います．目的の「ファイル」は，どこの棚にある何という名前のものかを指定しなければなりません．この指定を行うのが“ファイルディスクリプタ”と呼ばれるものです．ファイルディスクリプタは，棚にあたる〈入出力装置〉を指定する“デバイス名”と〈書類の名前〉にあたる“ファイル名”で構成されています．

ファイルをOPENした場合は，ファイル操作終了後必ずCLOSEを実行してください．特にディスクでは，ファイルをOPENしたまま(CLOSEせずに)，ディスクを差し替えると，新たに挿入したディスクのデータを壊してしまう危険性があります．

1 ファイルディスクリプタの諸元

1．ファイル番号とチャネル番号

N_{88}-BASIC/N_{88}-日本語BASICでは，入出力操作はファイルやチャネルの指定によって実行されます．ファイル番号とチャネル番号は同一の概念で，入出力のためのメモリ領域(バッファ)に対してつけた固有の番号です．

ファイル番号は補助記憶装置(フロッピーディスク，カセットテープなど)に対して用い，チャネル番号は入出力(通信)機器(プリンタ，スクリーン，RS-232C，キーボードなど)に対して用います．ファイル番号とチャネル番号は同一の概念ですから，重複して指定することは許されません．BASICを起動したとき指定する数は同時に扱うファイル(チャネル)の数で，この数の範囲内であれば何個でもファイル番号を指定することができます．

2．ファイルディスクリプタの構成

ファイルディスクリプタは各ファイル(デバイスも含む)を区別する名称で，次のような構成の文字列で表されます．

"〈デバイス名〉〔〈ファイル名〉〕"

ファイルディスクリプタは，必ずダブルクォーテーション(")で囲まなければなりません．デバイス名，ファイル名は場合によって省略することができます．デバイス名が省略できるのは，N_{88}-BASIC ROM Versionの場合カセットテープ1("CAS1:")，N_{88}-BASIC DISK Version, N_{88}-日本語BASICの場合フロッピーディスク1("1:")を指定するときです．

3．デバイス名

〈デバイス名〉は入出力機器の名称を表すもので，アルファベット4文字もしくはアルファベット3文字と数字1文字に"："(コロン)を付けたものを原則としています．ただし，フロッピーディスク装置については，数字(ドライブ番号を示す)1文字と"："の簡略形が用いられます．現在，N_{88}-BASIC/N_{88}-日本語BASICに定義されているデバイス名には次のものがあります．フロッピーディスク装置(n:)はN_{88}-BASIC DISK Version，またはN_{88}-日本語BASICでのみ使用できます．

デバイス名	機械名称	入力	出力	備考
LPT1: または**LPT:**	プリンタ	×	○	
SCRN:	スクリーン	×	○	
COM1: または**COM:**	RS 232Cポート1	○	○	
CAS1: または**CAS:**	カセットテープ1	○	○	N_{88}-BASIC ROM Version でのデフォルト，転送速度1200bps，
CAS2:	カセットテープ2	○	○	転送速度600bps
1:	フロッピーディスク1	○	○	N_{88}-BASIC DISK Version，N_{88}-日本語BASIC でのデフォルト．
2:	フロッピーディスク2	○	○	
3:	フロッピーディスク3	○	○	
4:	フロッピーディスク4	○	○	
5:	フロッピーディスク5	○	○	
6:	フロッピーディスク6	○	○	
7:	フロッピーディスク7	○	○	
8:	フロッピーディスク8	○	○	
KYBD:	キーボード	○	×	

注意 PC-8801MHで"**CAS1:**" "**CAS2:**"を使用するには，CMTインタフェースボード(オプション)を装着する必要があります．

4. オプション

〈オプション〉は，コミュニケーション(RS-232C)をデバイスとして指定したときのみ意味を持つもので，パリティチェックの有無など種々のモードを設定するのに用いられます．詳しくは「リファレンスマニュアル」のTERM文を参照してください．

"COM1:〔〈オプション〉〕"または"COM:〔〈オプション〉〕"

5. ファイル名

〈ファイル名〉とはプログラムファイルやデータファイルにつける名前で，ユーザが指定します．ファイル名を必要とするのは，カセット，フロッピーディスクなど補助記憶装置との入出力を行う場合で，その他の場合は省略することができます．ファイル名は次のような書式をとります．

〈ファイル名〉〔.〕〔〈拡張子〉〕

最初の〈ファイル名〉は6文字，ピリオドに続く〈拡張子〉は3文字までの文字列より構成されます．もし，それぞれの文字数がそれ以下の場合には，空いた部分に空白が埋められます．一般に〈ファイル名〉がファイルの名前を，〈拡張子〉がファイルの性質を表すようにしますが，ユーザによって自由に使うことができます．通常，ファイル名という場合は，〈拡張子〉まで含んだものを意味します．

〈ファイル名〉および〈拡張子〉の中には，キャラクタコード0～31(0H～1AH)の文字を含むことはできません．さらに，〈ファイル名〉の最

初の１文字にキャラクタコード255(FFH)の文字を使用することはできません．また，最初の〈ファイル名〉が６文字を超えて指定された場合，先頭から６文字が〈ファイル名〉，続く９文字までの３文字が〈拡張子〉と見なされ，それ以降の文字は無視されます．

2 ファイルを扱うための基本命令

ファイルを扱うための基本命令について，どのデバイスに対して有効なのかを表にしておきます．各命令の意味は「リファレンスマニュアル」を参照してください．

入出力命令＼デバイス	ディスクn:	CASn:	COM:	KYBD:	SCRN:	LPT:	備　考
BLOAD	○	×	○	×	×	×	○：使用できる． ×：使用できない． ＊：実際には何も行われない． ディスクn: は DISK Versionのみ可．
BSAVE	○	×	○	×	○	○	
CLOSE	○	○＊	○	○＊	○＊	○	
EOF	○	×	○	×	×	×	
FPOS	○	×	×	×	×	○	
GET	○	×	×	○	×	×	
INPUT#	○	○	○	○	×	×	
INPUT$	○	×	○	○	×	×	
LINE INPUT#	○	×	○	○	×	×	
LOAD	○	○	○	○	×	×	
LOC	○	×	○	○	×	×	
LOF	○	×	○	×	×	×	
OPEN	○	○＊	○	○＊	○＊	○	
PRINT# (PRINT# USING WRITE#)	○	○	○	×	○	○	
PUT	○	×	×	×	○	○	
SAVE	○	○	○	×	○	○	
WIDTH	×	×	○	×	×	○	

3 データファイルを扱った応用サンプル

ファイルディスクリプタは文字列ですから，文字変数を使うことができます．これにより，ファイル操作が便利に行えます．

下記のプログラムは，アスキー形式のデータファイルをCRTかプリンタに打ち出す例です．文字変数DEVICE$をスイッチとして用い，FILE$で入力ファイルを指定します．もし必要であれば，DEVICE$でカセットやRS-232Cポートを指定し，それらの装置のデバイス名("CAS1:"や"COM:"など)を付け加えることもできます．

〈アスキー形式のデータファイルをCRTかプリンタに打ち出すプログラムの例〉

```
100 ' FILE TYPE PROGRAM
110 '
120 INPUT "CRT (0) or PRINTER (1)";DEV
130 INPUT "FILE NAME";FILE$
140 '
150 IF DEV<0 OR 1<DEV THEN 120
160 IF DEV=0 THEN DEVICE$="SCRN:"
170 IF DEV=1 THEN DEVICE$="LPT1:"
180 '
190 OPEN FILE$ FOR INPUT AS #1
200 OPEN DEVICE$ FOR OUTPUT AS #2
210 '
220 WHILE NOT EOF(1)
230     LINE INPUT #1,A$
240     PRINT #2,A$
250 WEND
260 '
270 CLOSE
280 END
```

第14章

データファイルを作る

データファイルには，2つのタイプがあります．
1つは，データをファイルの先頭から順に読み書きするファイルで，「シーケンシャルファイル」といいます．もう1つは，データを任意の順に読み書きできるファイルで「ランダムファイル」といいます．
データファイルを作る際，どちらのタイプにするかは，プログラマーの自由です．ただ，一般的には高速処理を要求される場合やデータの更新が頻繁に行われる場合はランダムファイル，ファイル領域を小さくしたい場合やデータの更新があまりない場合はシーケンシャルファイルが使われます．
ただし，フロッピーディスク上には，この2つの形式のファイルが作れますが，カセットテープ上には，シーケンシャルファイルしか作れません．
本章では，フロッピーディスク上に作るシーケンシャルファイル，ランダムファイルについて説明したのち，資料としてカセットテープ上に作るファイルについて説明します．

> ファイルをOPENした場合は，ファイル操作終了後必ずCLOSEを実行してください．特にディスクでは，ファイルをOPENしたまま（CLOSEせずに），ディスクを差し替えると，新たに挿入したディスクのデータを壊してしまう危険性があります．
>
> またファイル操作中にエラーが発生した場合や 停止 を押してファイル操作を中断した場合は，CLOSEあるいはENDを実行してください．

1 シーケンシャルファイルとランダムファイルの比較

1. それぞれの特色

(1) **シーケンシャルファイル**

シーケンシャルファイルとは，データを連続的に記録するファイルの形式です．

このため，ファイルの使用効率がよいという特色があります．しかし，読み出す場合は，データの先頭から順に読み出すため，任意のデータの検索には向きません．また，修正，削除も部分的に行うことはできず，全データを更新する形でしか実現できません．

(2) **ランダムファイル**

ランダムファイルとは，シーケンシャルファイルの欠点である任意のデータへの素早いアクセスを可能としたファイルの形式です．

任意のレコード番号を指定することによって，それに対応するデータを取り出すことができます．また修正も，部分的に修正したいデータだけを変更することができます．

しかし，記録が1レコード単位で行われるため，利用効率は下がります．

2. 各ファイルで使用できるステートメントと関数

	シーケンシャルファイル	ランダムファイル
文	PRINT ＃ PRINT ＃ USING INPUT ＃ LINE INPUT ＃ WRITE ＃	FIELD LSET RSET PUT GET
関数	INPUT$ EOF LOC	CVI/CVS/CVD MKI$/MKS$/MKD$

2 シーケンシャルファイル

1. 一般的手順

シーケンシャルファイルのデータを読み書きするときは，一般に次の手順に従います．

(1) ファイルをオープンにする
(2) ファイルのデータを読み書きする
(3) ファイルをクローズにする

また，シーケンシャルファイルは，オープン時に次のいずれかのモードを指定します．

- OUTPUTモード： 新規にファイルを設定して，データを順次書き込む．
- APPENDモード： 既存のファイルの最後にデータを追加する．
- INPUTモード　： 既存のファイルの先頭からデータを順次読み込む．

2. シーケンシャルファイルの作成

(1) 作成手順

新たにシーケンシャルファイルを作り，データを読み書きするために必要な手順は次のようになります．

① 書き込みのためのファイルをオープンします．すでに同じ名前のファイルがあった場合には，以前のものを削除して新たに作りなおしますので，ご注意ください．

```
OPEN "TEST" FOR OUTPUT AS #1 ⏎
```

② WRITE#，PRINT#などを使って，ファイルにデータを書き込みます．

```
WRITE #1,A$,B ⏎
```

③ 必要なだけ②の作業を繰り返した後，ファイルをクローズして終了します．

```
CLOSE #1 ⏎
```

(2) サンプルプログラム

例として，キーボードから生徒のテストの点数を入力して，シーケンシャルファイル"TEST"を作るプログラムを示します．

```
100 ' ﾌﾟﾛｸﾞﾗﾑ 1
110  OPEN "TEST" FOR OUTPUT AS #1
120 *LOOP
130  INPUT "名前 ";N$
140  IF N$="end" THEN *FIN
150  INPUT "科目 ";K$
160  INPUT "得点 ";P
170  WRITE #1,N$,K$,P
180  PRINT
190 GOTO *LOOP
200 *FIN
210  CLOSE #1:END
```

注意　本章のサンプルプログラムはすべてN_{88}-日本語BASICを使用しています．N_{88}-BASICでサンプルプログラムを入力するときは，漢字やひらがな部分をカタカナで入力してください．

次のように入力し，このプログラムを実行してみましょう．

RUN ⏎

```
run
名前 ? 本田　卓也
科目 ? 理科
得点 ? 75

名前 ? 鈴木　蘭
科目 ? 理科
得点 ? 48

名前 ? 山葉　リサ
科目 ? 理科
得点 ? 47

名前 ? end
Ok
```

(3) WRITE#とPRINT#で作られるファイルの違い

式が2つ以上あった場合，WRITE#は，各式の値の間は自動的にコンマ(,)で区切られます．また，文字列はダブルクォーテーション("　")で囲まれて出力されます(例1)．

PRINT#では，各式の値の間はコンマ(,)で区切られず，連続して出力されます(例2)．また，文字列をダブルクォーテーション("　")で囲むということもしません．このため，たとえば文字列を続けて出力すると，それらは連続した1つの文字列となります．このため，WRITE#の場合のように，コンマ(,)で各式の値の間を区切るには， 例3 のように各式の間に"，"を出力する必要があります．

使用するステートメントと作成されるファイルの対応を図示すれば，次のとおりです．

例1　WRITE #1, N$, K$, P

```
"本田　卓也","理科",75CRLF|"鈴木
蘭","理科",48CRLF|"山葉　リサ","理
科",47CRLF
```

例2　PRINT #1, N$; K$; P

```
本田　卓也理科75CRLF|鈴木　蘭理科4
8CRLF|山葉　リサ理科47CRLF
```

例3　PRINT #1, N$; ","; K$; ","; P

```
本田　卓也,理科,75CRLF|鈴木　蘭,理
科,48CRLF|山葉　リサ,理科,47CRLF
```

またPRINT#では，PRINT# USINGを使って，編集されたデータを書き込むこともできます．たとえば，

```
100 PRINT #1,USING"&              &";N$;
110 PRINT #1,USING"&            &";K$;
120 PRINT #1,USING"###";P
```

```
本田　卓也　　　理科　　　　75CRLF|鈴木
　蘭　　　　　理科　　　　48CRLF|山葉　リ
サ　　　　理科　　　　47CRLF
```

3. シーケンシャルファイルからの読み込み

(1)　**一般的手順**

①　読み込みのためのファイルをオープンします．

OPEN "TEST" FOR INPUT AS #1

②　INPUT#, LINE INPUT#などを使って，ファイルからデータを読み込みます．

INPUT #1, A$, B

③　必要なだけ②の作業を繰り返した後，ファイルをクローズして終了します．

CLOSE #1

(2)　**サンプルプログラム**

次にプログラム2として，プログラム1で作ったファイル"TEST"をアクセスし，50点以下の生徒を全部表示する例を下に示します．

```
100 ' ﾌﾟﾛｸﾞﾗﾑ 2
110  OPEN "TEST" FOR INPUT AS #1
120 *LOOP
130  INPUT #1,N$,K$,P
140  IF P>=50 THEN *LOOP
150  PRINT N$,K$,P
160 GOTO *LOOP
```

これを実行してみると，次のようになります．

```
run
鈴木　蘭　　　理科　　　　　48
山葉　リサ　　理科　　　　　47
Input past end in 130
Ok
```

プログラム2では，ファイル中のデータを最初から順番に読み込んでいきます．そして，ファイルの最後でもう読み込むデータがなくなると，130行で"Input past end"エラーが起こります．

これを避けるためには，125行に下の1行を挿入します．

125 IF EOF(1) THEN *FIN

この125行でファイルの終わりをチェックし，次の3行のように処理します．

170 *FIN
180　CLOSE #1
190 END

(3) INPUT＃とLINE INPUT＃の違い

両者の違いは，何をデータの区切り符号とするかが違います．

INPUT＃は，コンマ，スペース，キャリッジリターン，ラインフィードなどを区切り符号とします．

LINE INPUT＃は，キャリッジリターン，ラインフィードのみを区切り符号とします．このため，コンマやダブルクォーテーションもデータとして読み出すことができます．

4. シーケンシャルファイルへのデータの追加

(1) 一般的手順

フロッピーディスク上にシーケンシャルファイルがすでに作られていて，その後にデータを追加したい場合には，次のような手順で行います．

① 追加するためのファイルをアペンドモードでオープンします．

OPEN "TEST" FOR APPEND AS ＃1 ⏎

② WRITE＃，PRINT＃などを使って，ファイルに追加したいデータを書き込みます．

WRITE ＃1，N$；P ⏎

③ 必要なだけ②の作業を繰り返した後，ファイルをクローズして終了します．

CLOSE ＃1 ⏎

(2) サンプルプログラム

例として，プログラム1で作ったファイル"TEST"にデータを追加する例をプログラム3として示します．

```
100 ' ﾌﾟﾛｸﾞﾗﾑ 3
110  OPEN "TEST" FOR APPEND AS #1
120 *LOOP
130  INPUT "名前 ";N$
140  IF N$="end" THEN *FIN
150  INPUT "科目 ";K$
160  INPUT "得点 ";P
170  WRITE #1,N$;K$;P
180  PRINT
190 GOTO *LOOP
200 *FIN
210  CLOSE #1:END
```

これを実行してみると，次のようになります．

```
run
名前 ? 本田　卓也
科目 ? 国語
得点 ? 46

名前 ? 鈴木　蘭
科目 ? 国語
得点 ? 74

名前 ? 山葉　リサ
科目 ? 国語
得点 ? 8Ø

名前 ? end
Ok
```

これは，プログラム1の110行のOUTPUTをAPPENDに代えただけです．このプログラムでデータを追加したファイルに対して，プログラム2を実行してチェックすることができます．

5. シーケンシャルファイルのデータの修正

シーケンシャルファイルは，そのままではファイルのデータを修正することができません．現在のファイルから読み込んで修正したデータを新しい別のシーケンシャルファイルへ書き込んでいく作業を行わなければなりません．下にプログラム4として，プログラム1や3で作ったファイルの修正を行う例を示します．

プログラム4では，ファイル"TEST"から読み込んだデータに修正がない場合は，そのまま新しいファイル"test"へ書き込みます．もし修正がある場合は，260～280行で新しいデータを入力し，"test"へ書き込みます．削除する場合には，書き込まずに次のデータを"TEST"から読み込みます．修正が終了した場合には，"TEST"から読み込んだデータをそのまま"test"へ書き込む作業を"TEST"のファイルエンドまで繰り返します．

最後に，古いファイル"TEST"を消去し，"test"を"TEST"に名前を変更して終了します．

```
100 ' ﾌﾟﾛｸﾞﾗﾑ 4
110  OPEN "TEST" FOR INPUT AS #1
120  OPEN "test" FOR OUTPUT AS #2
130 *LOOP
140  IF EOF(1) THEN *FIN
150  INPUT #1,N$,K$,P
160  IF FLG=1 THEN *WR
170  PRINT N$,K$,P
180 *QUESTION
190  PRINT "変更がありますか ";
200  INPUT "（y 又は n 又は d 又は e ）";A$
210  A$=LEFT$(A$,1)
220  IF A$="e" THEN FLG=1:GOTO *WA
230  IF A$="d" THEN *LOOP
240  IF A$="n" THEN PRINT:GOTO *WR
250  IF A$<>"y" THEN *QUESTION
260  INPUT "名前 ";N$
270  INPUT "科目 ";K$
280  INPUT "得点 ";P
290  PRINT
300 *WR
310  WRITE #2,N$;K$;P
320 GOTO *LOOP
330 *FIN
340  CLOSE #1
350  CLOSE #2
360  KILL "TEST"
370  NAME "test" AS "TEST"
380 END
```

3 ランダムファイル

1. 一般的手段

ランダムファイルのデータを読み書きするときは，一般に次の手順に従います．

(1) ファイルをオープンする
(2) バッファのフィールドデータを定義する
(3) バッファを経由しデータを読み書きする
(4) ファイルをクローズする

上の(2)でいうバッファ(ファイルバッファともいいます)は，メモリ上に確保された256バイトの領域で，このバッファを介し，256バイト単位でデータのやりとりが行われます．

2. ランダムファイルの作成

(1) 作成手順

① ファイルをオープンする．

```
OPEN "TEST" AS #1
```

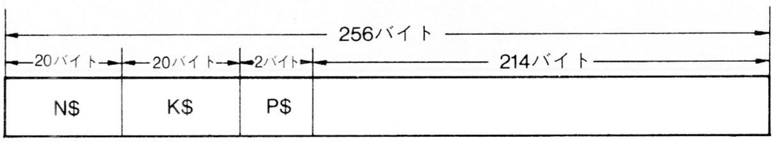
ランダムファイルをオープンするときは，モードの指定は不要です．

なお，OPEN文で指定したファイルがディスク上にないと新たにファイルがつくられますが，ファイルがディスク上にあれば，既存のファイルをオープンします．

② フィールドを定義する．

```
FIELD #1, 20 AS N$, 20 AS K$, 2 AS P$
```

ファイルバッファの中に文字変数の領域をバイト単位で割りあてるのがFIELD文です．各文字変数の領域の合計は256バイト以下でなければなりません．上の用例であれば，バッファは次のように割りあてられます．



なお，FIELD文中で使った文字変数を，INPUT文や代入文の左辺で使用することはできません．

③ LSET, RSETを用い，バッファにデータをセットする．

```
LSET N$=A$ ⏎
```

バッファにデータをセットするのは，代入文ではできません．LSETまたはRSETを使わなければなりません．LSETは左づめ，RSETは右づめにして，それぞれバッファにデータをセットします．また，バッファにセットするデータは文字列でなければなりません．このため，データが数値である場合には，MKI$，MKS$，MKD$を使って，それぞれ整数値，単精度実数値，倍精度実数値を2バイト，4バイト，8バイトの文字に変換してからバッファへ転送します．

```
LSET P$=MKI$(P%) ⏎
```

④ バッファのデータをディスクに書き込む．

```
PUT #1,R ⏎
       └─ レコード番号
```

Rはレコード番号です．

ランダムファイルでは，レコードの長さは256バイトで，各レコードには1から順に番号が振られます．この番号のことをレコード番号と呼び，ランダムファイルのレコードを読み書きするときには，必ずこのレコード番号を指定しなければなりません．

⑤ ファイルをクローズする．

```
CLOSE #1 ⏎
```

(2) サンプルプログラム

次のプログラムは，先のシーケンシャルファイルのサンプルプログラムと同じ内容を，ランダムファイルとして作成するものです．

```
100  ' RAMDAM FILE / DATA WRITE
110 R=1
120 OPEN "test" AS #1
130   FIELD #1,20 AS N$,20 AS K$,2 AS P$
140 *LOOP
150   INPUT "名前 ";A$
160   IF A$="end" THEN *FIN
170   LSET N$=A$
180   INPUT "科目 ";A$
190   LSET K$=A$
200   INPUT "得点 ";P
210   A$=MKI$(P)
220   LSET P$=A$
230   PUT #1,R
240   R=R+1
250   PRINT
260   GOTO *LOOP
270 *FIN
280 CLOSE #1
290 END
```

では，実行してみましょう．

```
run
名前 ? 本田　卓也
科目 ? 理科
得点 ? 75

名前 ? 鈴木　蘭
科目 ? 理科
得点 ? 48

名前 ? 山葉　リサ
科目 ? 理科
得点 ? 47

名前 ? end
Ok
```

3. ランダムファイルの読み込み

(1) 読み込み手順

① ファイルをオープンする．

OPEN "TEST" AS #1 ⏎

モード指定は不要です．

② フィールドを定義する．

FIELD#1, 20 AS N$, 20 AS K$, 2 AS P$ ⏎

作成したときと同じに定義します．

③ ディスクのデータをバッファに読み込む．

GET #1, R ⏎

読み込む場合，読み込みたいレコードの番号を指定します．すでに作成されているレコード数を越える指定はエラーとなります．

④ バッファのデータを取り出す．

A$=N$ ⏎

バッファのデータを取り出すには代入文を使います．単に表示させるだけなら，PRINT 文で直接指定できます．

PRINT N$; CVI(P$) ⏎

GET#によって読み込んだデータが数値である場合には，CVI, CVS, CVDを使って，それぞれ文字列から整数，単精度実数値，倍精度実数値へ変換します．

P=CVI(P$) ⏎

⑤ ファイルをクローズする．

CLOSE #1 ⏎

(2) サンプルプログラム

```
100  '  RANDOM FILE / Data read
120 OPEN "test" AS #1
130   FIELD#1,20 AS N$,20 AS K$,2 AS P$
140   MAX=LOF(1)
150   FOR R=1 TO MAX
160     GET#1,R
170     PRINT N$;K$;CVI(P$)
180   NEXT R
190 CLOSE #1
200 END
```

これを実行すると，次のようになります．

```
run
本田　卓也　　　　　理科　　　　　　75
鈴木　蘭　　　　　　理科　　　　　　48
山葉　リサ　　　　　理科　　　　　　47
Ok
```

4. ランダムファイルによる得点検索プログラム

下のプログラム5は，生徒番号によってランダムに読み書きするものです．新しくファイルを作る場合は，RUN実行後に"ファイル名"を入力したら，次に必ず"ファイルの初期化"を実行してください．

```
100 ' ﾌﾟﾛｸﾞﾗﾑ 5
110  CLS:INPUT "ファイル名  ";F$
120  OPEN F$ AS #1
130  FIELD #1,1 AS ID$,20 AS N$,20 AS K$,2 AS P$
140 *FUNC
150  CLS
160  PRINT "  1：データ作成，変更"
170  PRINT "  2：データ参照"
180  PRINT "  3：点数による検索"
190  PRINT "  4：終了"
200  PRINT "999：ファイルの初期化"
210  INPUT "  選んでください";F
220   IF F=999 THEN *FINI
230  ON F GOTO *HENKO,*SANSHO,*KENSAKU,*FIN
240 GOTO *FUNC
250 *SANSHO
260  SFLG=-1
270 GOTO *HKLOOP
280 *HENKO
290  SFLG=0
300 *HKLOOP
310  GET #1,1
320  MAX=CVI(P$)
330  PRINT "生徒番号（1-";USING "###";MAX-1;
340  INPUT " 又は 0:終了）";NUM%
350   IF NUM%=0 THEN *FUNC
360  NUM%=NUM%+1
370  GET #1,NUM%
380   IF ID$=CHR$(255) AND SFLG=0 THEN *QUESTION
390   IF ID$=CHR$(255) AND SFLG THEN *HFLOOP
400  P%=CVI(P$)
410  PRINT
420  PRINT "名前 ：";N$
430  PRINT "科目 ：";K$
440  PRINT "得点 ：";P%
450  PRINT
460   IF SFLG THEN *HKLOOP
470 *QUESTION
480  INPUT "名前 ";A$
490   IF A$<>"" THEN LSET N$=A$
500  INPUT "科目 ";A$
510   IF A$<>"" THEN LSET K$=A$
520  INPUT "得点 ";A$
530   IF A$<>"" THEN LSET P$=MKI$(VAL(A$))
540  PRINT
550  LSET ID$="U"
560  PUT #1,NUM%
570 GOTO *HENKO
580 *KENSAKU
590  GET#1,1
600  MAX=CVI(P$)
610  INPUT "何点以下で検索しますか ";AKA
```

```
620  PRINT
630  FOR NUM%=2 TO MAX
640   GET #1,NUM%
650    IF ID$=CHR$(255) THEN *KLOOP
660    IF CVI(P$)>AKA THEN *KLOOP
670   PRINT USING "### ";NUM%-1;
680  PRINT N$;K$;USING " ####";CVI(P$)
690 *KLOOP
700  NEXT NUM%
710  PRINT
720  PRINT "検索終了. キーをたたいてください"
730 *LOOP
740 A$=INKEY$:IF A$="" THEN *LOOP
750 GOTO *FUNC
760 *FINI
770  INPUT "最大の生徒番号は ";MAX
780  INPUT "ファイルを初期化します (yes 又は no) ";A$
790   IF A$<>"yes" THEN *FUNC
800  FOR NUM%=1 TO MAX+1
810   LSET ID$=CHR$(255)
820   PUT #1,NUM%
830  NEXT NUM%
840  LSET P$=MKI$(MAX+1)
850  PUT #1,1
860 GOTO *FUNC
870 *FIN
880  CLOSE #1
890 END
```

資料 4 カセットテープ上にデータファイルを作る

注意 ● PC-8801MHには，カセットテープ用のインタフェースが装備されていません．

PC-8801MHでカセットテープを使用するには，オプションのCMTインタフェースボードが必要になります．

PC-8801FHの場合は，オプションなしに，以後の操作が行えます．

● カセットテープレコーダを使用して，データの書き込み，読み込みを行う場合は，リモート端子(黒)を接続します．

リモート端子を接続しますと，テープレコーダのON/OFFは本機がコントロールします．ですから，PLAYボタンを押していても，本機でカセットテープを使用する命令が実行されなければカセットテープは回りません．この機能によって，必要な間だけカセットテープを回すことができ，カセットテープのむだづかいが防げ，また手間もかかりません．

リモート端子を接続しなかった場合は，テープレコーダのスタート，ストップはすべて自分でやる必要があります．ですから，カセットを使用する命令を実行する前にPLAYボタンを押し，終わるとSTOPボタンを押すという操作が必要になります．

以後は，リモート端子が接続されているとして説明します．

1. データの書き込み

(1) キーボードから下のプログラムを入力してください．

```
100 INPUT A,B,C,D
110 PRINT "カセットテープをセットし、";
120 PRINT "録音ボタンとＰＬＡＹボタンを";
130 PRINT "押してください。"
140 INPUT "Ok";A$
150 OPEN "cas:suchi" FOR OUTPUT AS #1
160 PRINT #1,A,B
170 PRINT #1,C,D
180 CLOSE #1
190 END
```

このプログラムを簡単に説明しましょう．

① キーボードから数値を入力します(100行)．

② カセットテープをセットし，録音ボタンとPLAYボタンを押して ⏎ を入力します(110行〜140行)．

③ ファイルのオープンを行います．OPEN文が実行されるとカセットテープが割り当てられ，データを書き込むための準備が整います(140行)．

④ PRINT＃文を用いてデータを書き込みます(160行，170行)．

⑤ データの書き込みが終わったら，CLOSE文でファイルを閉じておきます(180行)．

(2) このプログラムを実行させます．

run ⏎

(3) すると，画面に"?"を表示します．これは，行番号100のINPUT文がキーボードからの入力を要求しているのです．

1,2,3,4

と入力します．

(4) 今度は次のメッセージが表示されます．

```
Set tape, then push PLAY and REC
Ok?
```

(5) そこで，データを入力してもよいカセットテープをテープレコーダにセットします．このとき，テープカウンタを0にしておけば，後でどこからデータが入っているか一目でわかり，とても便利です．

(6) カセットテープのセットが終わったら，テープレコーダの録音ボタンとPLAYボタンを押します．これらの準備が終わったら，⏎を入力します．

(7) ⏎を入力すると，すぐにカセットテープが回り始め，しばらくしてディスプレイに"Ok"と出力し，カセットテープは止まります．

これで，キーボードから入力した数値がカセットテープに書き込まれました．

2. データの読み込み

先ほどテープに書き込んだデータを，今度は読み込んでみましょう．

(1) 下のプログラムを入力します．

```
100 PRINT "ＰＬＡＹボタンを押してください。"
110 OPEN "cas:suchi" FOR INPUT AS #1
120 INPUT #1,A,B
130 INPUT #1,C,D
140 PRINT A,B,C,D
150 CLOSE #1
160 END
```

プログラムの意味は次のとおりです．

① 画面に"PLAYボタンを押してください。"と表示し，PLAYボタンを押すことを促します(100行)．

② ファイルをオープンし，カセットテープからデータを入力するためのバッファを割り当てます(110行)．

③ INPUT#文を用いてデータを読み込みます(120行，130行)．

④ データの読み込みが終わったら，CLOSE文でファイルを閉じます(150行)．

(2) プログラムが入力できたら，データの書き込みを始めたところまでカセットテープを巻き戻します．

(3) カセットテープの準備ができたら実行させます．

run ⏎

(4) "PLAYボタンを押してください。"というメッセージが現れたら，PLAYボタンを押します．カセットテープが回り始めてしばらくすると，読み込まれたデータがディスプレイに表示されます．

3. データの型と数の一致

カセットテープにデータを書き込む，あるいは読み込む場合の注意事項として，書き込むときと読み込むときの〈データの型と数の一致〉があります．文字としてデータを書き込んだのに，それを数値データとして読み込むことはできません．また，1つのPRINT文で4つのデータを書き込んでいるのに，INPUT文では2つしか読み込まないというわけにもいきません．ただし，データの型と数が一致していれば，読み込んだデータを代入する変数は，書き込んだときに使った変数と異なっていてもかまいません．

良い例1

・書き込むとき

```
10 OPEN "cas:sample" FOR OUTPUT AS #1
        ⋮
100 PRINT #1,A,B,C
        ⋮
500 CLOSE #1
```

・読み込むとき

```
10 OPEN "cas:sample" FOR INPUT AS #1
        ⋮
150 INPUT #1,X,Y,Z
        ⋮
300 CLOSE #1
```

データの型(数値)，数(３つずつ)が一致しているのでOKですね．

良い例２

・書き込むとき

```
10 OPEN "cas:sample" FOR OUTPUT AS #1
        ⋮
100 PRINT #1,YAMA$,KAWA$
        ⋮
500 CLOSE #1
```

・読み込むとき

```
10 OPEN "cas:sample" FOR INPUT AS #1
        ⋮
200 INPUT #1,SORA$,UMI$
        ⋮
600 CLOSE #1
```

これも，データの型(文字)，数(２つずつ)が一致しているのでOKです．

悪い例

・書き込むとき

```
10 OPEN "cas:sample" FOR OUTPUT AS #1
        ⋮
100 PRINT #1,A,B,C,D
        ⋮
700 CLOSE #1
```

・読み込むとき

```
10 OPEN "cas:sample" FOR INPUT AS #1
        ⋮
200 INPUT #1,A,B
210 INPUT #1,C,D
        ⋮
800 CLOSE #1
```

データの型(数値)は一致していますが，１つのPRINT文で４つのデータを書き込んだのに対し，読み込むときは一度に２個しか扱っていません．この場合，読み込むことはできません．

4. ファイル名について

カセットテープ上にデータファイルをオープンして入出力を行う場合は，フロッピーディスクの場合と同じようにファイル名をつけることが可能ですが，フロッピーディスクのようにファイル名によってファイルを識別することはできません．

したがって，

OPEN "CAS : A" FOR OUTPUT AS #1 ↵

としてオープンし，データを書き込んだファイルを，

OPEN "CAS : B" FOR INPUT AS #1 ↵

としてオープンし，データを読み出すこともできます．

5. カセットテープの転送速度

N_{88}-BASICでカセットテープを使用する場合に，「600bps」と「1200bps」の2つの転送速度を使用します．この指定は，デバイス名を指定すると，N_{88}-BASICが自動的に設定してくれます．

デバイス名	転送速度
cas: (cas1:)	1200bps
cas2:	600bps

同じ長さのプログラムやデータをセーブしたりロードする場合，1200bpsですと600bpsの半分の時間で実行できます．しかし，600bpsでセーブしたプログラムやデータを1200bpsでロードすることはできません．また，逆に1200bpsでセーブしたプログラムやデータを600bpsでロードすることもできません．ですから，プログラムやデータをセーブする場合は，どちらの転送速度でセーブしたのかを記録しておいてください．

注意 同じテープレコーダでセーブ，ロードを行う場合は，「1200bps」で行う方が速く実行できますが，セーブしたテープレコーダとロードするテープレコーダが異なる場合に，「1200bps」ですと，モータの回転速度の誤差などからエラーが発生する可能性があります．

このような場合は「600bps」でセーブ，ロードを行ってください．

第15章

フロッピーディスクとファイルの構造

N88-BASIC DISK Version／N88-日本語BASICにおけるフロッピーディスクのファイル構造について解説します．
BASICでディスクファイルを扱うときは，すべてファイル名で行います．したがって，フロッピーディスクを直接管理するサーフェス番号，トラック番号，セクタ番号を使う必要はほとんどありません．なぜならば，ファイル名を実際のフロッピーディスクのサーフェス番号，トラック番号，セクタ番号に変換する仕事はBASICが自動的に行ってくれるからです．
したがって，BASICでファイルを扱われる方は以下の説明を読む必要はありません．DSKI$関数，DSKO$文や機械語モニタを使う必要のある方のみお読みください．

> DSKO$文は，ディスク上に直接データを書き込みます．不用意に使用すると，ディスクのデータ構造を破壊し，通常のファイル操作が不可能になる場合もありますので，使用に当たっては十分な注意が必要です．

1 フロッピーディスク

1. 使用するディスクの種類

N_{88}-BASIC DISK Version/N_{88}-日本語BASICで通常使用するフロッピーディスクは次のとおりです．

(1) PC-8801FHの場合

- 5.25インチ両面倍密度ミニフロッピーディスク(2Dと呼ばれます)
- 8インチ両面倍密度フロッピーディスク(2Dと呼ばれます)

(2) PC-8801MHの場合

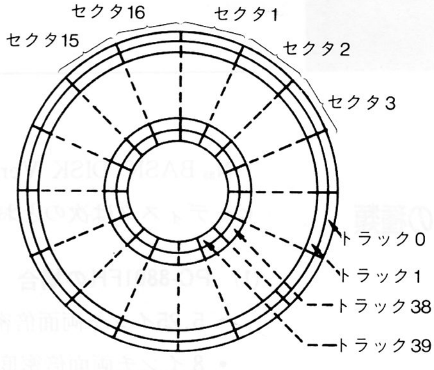
- 5.25インチ両面倍密度ミニフロッピーディスク(2Dと呼ばれます)
- 5.25インチ両面高密度ミニフロッピーディスク(2HDと呼ばれます)
- 8インチ両面倍密度フロッピーディスク(2Dと呼ばれます)

2. ディスクの番地指定

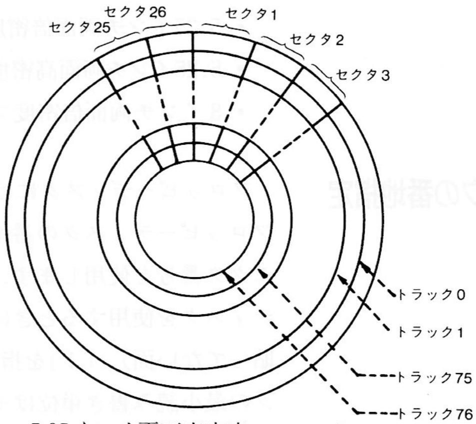
フロッピーディスクには，メモリと同じように番地がついています．フロッピーディスクの番地指定には，サーフェス番号，トラック番号，セクタ番号を使用します．サーフェス番号は，両面倍密度のフロッピーディスクを使用するときに必要です．「0」を指定すると表面(ラベルの貼ってない面)，「1」を指定すると裏面になります．フロッピーディスクの最小読み書き単位はセクタで，1セクタは256バイトから成ります．

3. ディスクの構成

BASICで使用する3種類のフロッピーディスクの構成は下図のようになっています．5.25インチ両面高密度ミニフロッピーディスクと，8インチ両面倍密度フロッピーディスクは全く同じ構成になっています．



5.25インチ両面倍密度ミニフロッピーディスク



5.25インチ両面高密度フロッピーディスク
8インチ両面倍密度フロッピーディスク

	サーフェス番号	トラック数	トラック番号	セクタ数	セクタ番号
5.25インチ両面倍密度	0か1	片面40 (80)	0～39	16	1～16
5.25インチ両面高密度と8インチ両面倍密度	0か1	片面77 (154)	0～76	26	1～26

2 トラックの割り当て

1. システムディスクとデータディスク

システムディスクとは，ディスクコードを含んだフロッピーディスクのことで，N_{88}-BASIC DISK Version，およびN_{88}-日本語BASICを起動させる際に必要なものです．ディスクコードとは，BASICの拡張部分(フロッピーディスク関係の命令のサポート)のことです．それとは逆に，ディスクコードを含まないフロッピーディスクのことをデータディスクと呼びます．データディスクではN_{88}-BASIC DISK Version，およびN_{88}-日本語BASICを起動させることはできません．

プログラムやデータは，システムディスクとデータディスクの両方に記憶させることができます．

2. トラックの割り当て

(1) 5.25インチ両面倍密度ミニフロッピーディスク

N_{88}-BASICシステムディスク

サーフェス	トラック		セクタ		内容
	10進	16進	10進	16進	
0	0	0H	1	1H	IPL
	0	0H	2～16	2H～10H	ディスクコード
	1	1H	すべて	すべて	ディスクコード
	2～39	2H～27H	すべて	すべて	ユーザ領域
1	0～1	0H～1H	すべて	すべて	ディスクコード
	2～17	2H～11H	すべて	すべて	ユーザ領域
	18	12H	1～12	1H～CH	ディレクトリ
	18	12H	13	DH	ID
	18	12H	14～16	EH～10H	FAT
	19～39	13H～27H	すべて	すべて	ユーザ領域

N_{88}-日本語BASICシステムディスク

サーフェス	トラック		セクタ		内容
	10進	16進	10進	16進	
0	0	0H	1	1H	IPL
	0	0H	2～16	2H～10H	ディスクコード
	1～5	1H～5H	すべて	すべて	ディスクコード
	6～31	6H～1FH	すべて	すべて	辞書
	32～39	20H～27H	すべて	すべて	ユーザ領域
1	0～4	0H～4H	すべて	すべて	ディスクコード
	5～17	5H～11H	すべて	すべて	辞書
	18	12H	1～12	1H～CH	ディレクトリ
	18	12H	13	DH	ID
	18	12H	14～16	EH～10H	FAT
	19～30	13H～1EH	すべて	すべて	辞書
	31～39	1FH～27H	すべて	すべて	ユーザ領域

データディスク

サーフェス	トラック		セクタ	内容
	10進	16進		
0	0～39	0H～27H	すべて	ユーザ領域
1	0～17	0H～11H	すべて	ユーザ領域
	18	12H	すべて	システムディスクと同じ
	19～39	13H～27H	すべて	ユーザ領域

(2)　5.25インチ両面高密度ミニフロッピーディスク

N_{88}-BASICシステムディスク

サーフェス	トラック		セクタ		内　容
	10進	16進	10進	16進	
0	0	0H	1~2	1H~2H	IPL
	0	0H	3~26	3H~1AH	未使用
	1	1H	1~2	1H~2H	IPL
	1	1H	3~26	3H~1AH	ディスクコード
	2	2H	すべて	すべて	ディスクコード
	3~34	3H~22H	すべて	すべて	ユーザ領域
	35	23H	1~22	1H~16H	ディレクトリ
	35	23H	23	17H	ID
	35	23H	24~26	18H~1AH	FAT
	36~76	24H~4CH	すべて	すべて	ユーザ領域
1	0	0H	すべて	すべて	未使用
	1	1H	すべて	すべて	ディスクコード
	2~76	2H~4CH	すべて	すべて	ユーザ領域

N_{88}-日本語BASICシステムディスク

サーフェス	トラック		セクタ		内　容
	10進	16進	10進	16進	
0	0	0H	1~2	1H~2H	IPL
	0	0H	3~26	3H~1AH	未使用
	1~5	1H~5H	すべて	すべて	ディスクコード
	6~31	6H~1FH	すべて	すべて	辞書
	32~34	20H~22H	すべて	すべて	ユーザ領域
	35	23H	1~22	1H~16H	ディレクトリ
	35	23H	23	17H	ID
	35	23H	24~26	18H~1AH	FAT
	36~76	24H~4CH	すべて	すべて	ユーザ領域
1	0	0H	すべて	すべて	未使用
	1~5	1H~5H	すべて	すべて	ディスクコード
	6~31	6H~1FH	すべて	すべて	辞書
	32~76	20H~4CH	すべて	すべて	ユーザ領域

データディスク

サーフェス	トラック		セクタ		内　容
	10進	16進	10進	16進	
0	0	0H	すべて	すべて	未使用
	1~34	1H~22H	すべて	すべて	ユーザ領域
	35	23H	1~22	1H~16H	ディレクトリ
	35	23H	23	17H	ID
	35	23H	24~26	18H~1AH	FAT
	36~76	24H~4CH	すべて	すべて	ユーザ領域
1	0	0	すべて	すべて	未使用
	1~76	1H~4CH	すべて	すべて	ユーザ領域

注意　5.25インチ両面高密度ミニフロッピーディスクでは，サーフェス0，トラック0は1セクタ128バイトの単密度になっているため，BASICでは使用できません．

(3) 8インチ両面倍密度フロッピーディスク

N_{88}-BASICシステムディスク

サーフェス	トラック		セクタ		内　容
	10進	16進	10進	16 進	
0	0	0H	すべて	すべて	未使用
	1	1H	1	1H	IPL
	1	1H	2～26	2H～1AH	ディスクコード
	2	2H	すべて	すべて	ディスクコード
	3～34	3H～22H	すべて	すべて	ユーザ領域
	35	23H	1～22	1H～16H	ディレクトリ
	35	23H	23	17H	ID
	35	23H	24～26	18H～1AH	FAT
	36～76	24H～4CH	すべて	すべて	ユーザ領域
1	0	0H	すべて	すべて	未使用
	1	1H	すべて	すべて	ディスクコード
	2～76	2H～4CH	すべて	すべて	ユーザ領域

データディスク

サーフェス	トラック		セクタ	内　　容
	10進	16進		
0	0	0H	すべて	未使用
	1～34	1H～22H	すべて	ユーザ領域
	35	23H	すべて	システムディスクと同じ
	36～76	24H～4CH	すべて	ユーザ領域
1	0	0H	すべて	未使用
	1～76	1H～4CH	すべて	ユーザ領域

注意 8インチ両面倍密度フロッピーディスクでは，サーフェス0，トラック0は1セクタ128バイトの単密度になっているため，BASICでは使用できません．

3 クラスタ

フロッピーディスクで読み書き可能な最小単位はセクタですが，BASICが管理する最小単位はクラスタと呼びます．

5.25インチ両面倍密度ミニフロッピーディスクは8セクタ，5.25インチ両面高密度ミニフロッピーディスクと8インチ両面倍密度フロッピーディスクは26セクタをひとまとめにして1クラスタとします．クラスタとサーフェス，トラック，セクタの関係は下の表のようになっています．

5.25インチ両面倍密度ミニフロッピーディスクの場合

クラスタ		サーフェス		トラック		セクタ	
10進	16進	10進	16進	10進	16進	10進	16進
0	0H	0	0H	0	0H	1～8	1H～ 8H
1	1H	0	0H	0	0H	9～16	9H～10H
2	2H	1	1H	0	0H	1～8	1H～ 8H
3	3H	1	1H	0	0H	9～16	9H～10H
4	4H	0	0H	1	1H	1～8	1H～ 8H
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
158	9EH	1	1H	39	27H	1～8	1H～ 8H
159	9FH	1	1H	39	27H	9～16	9H～10H

5.25インチ両面高密度ミニフロッピーディスク／
8インチ両面倍密度フロッピーディスクの場合

クラスタ		サーフェス		トラック		セクタ	
10進	16進	10進	16進	10進	16進	10進	16進
0	0H	0	0H	0	0H	1～26	1H～1AH
1	1H	1	1H	0	0H	1～26	1H～1AH
2	2H	0	0H	1	1H	1～26	1H～1AH
3	3H	1	1H	1	1H	1～26	1H～1AH
4	4H	0	0H	2	2H	1～26	1H～1AH
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
152	98H	0	0H	76	4CH	1～26	1H～1AH
153	99H	1	1H	76	4CH	1～26	1H～1AH

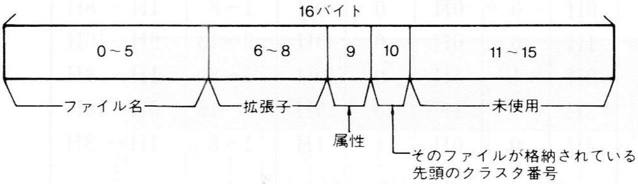
4 ディレクトリ，FAT

ユーザから指定されたファイル名を，フロッピーディスク上のアドレス(サーフェス番号，トラック番号，セクタ番号)に変換するために，BASICでは２つの表を管理しています．ディレクトリとFAT(File Allocation Tableの略)がそれです．

ディレクトリは，そのディスクに格納されているファイルのファイル名，属性，格納されている先頭の場所の表です．各ディレクトリは16バイトから成り，下記のような内容を持っています．

ファイル名の先頭バイト(０バイト目)の内容がFF(16進数)のとき，そのディレクトリが使用されていないことを示します．00のとき，削除(KILL)されたファイルであることを示します．

属性は下のようになっています．



バイト 10進	バイト 16進	内容
0～5	0H～5H	ファイル名
6～8	6H～8H	拡張子
9	9 H	属性
10	AH	そのファイルが格納されている先頭のクラスタ番号
11～15	BH～FH	未使用

属　　性

値(16進数)	意　　味
00H	アスキー形式
01H	機械語ファイル
10H	アスキー形式，書き込み禁止
40H	アスキー形式，リードアフターライト
80H	非アスキー形式，
90H	非アスキー形式，書き込み禁止
A0H	非アスキー形式，暗号化
C0H	非アスキー形式，リードアフターライト

FATは，各クラスタの使用状態を示す表です．FATは3セクタにわたって書き込まれており，3セクタとも全く同じ内容が書かれています．フロッピーディスクをアクセスしたときに，FATの内容(3セクタ)がメモリにコピーされます．1クラスタにつき1バイトでそのクラスタの使用状態を表します．5.25インチ両面倍密度ミニフロッピーディスクの場合，各セクタの0～159バイトがクラスタ番号0～159にそれぞれ対応しています．5.25インチ両面高密度ミニフロッピーディスクと8インチ両面倍密度フロッピーディスクの場合，各セクタの0～153バイト目がクラスタ番号0～153にそれぞれ対応しています．そして，各バイトの持っている値が各クラスタの使用状態を示しています．

表の説明で「連続したクラスタ」といいましたが，これは物理的(地理的)に連続したクラスタという意味ではありません．BASICで管理されるファイルは一般に，物理的に飛び飛びのクラスタをつないで構成されています．

バイトのデータ(16進)	クラスタの使用状態
5.25インチ両面倍密度のとき 0H～9FH 5.25インチ両面高密度と8インチ両面倍密度のとき 0H～99H	使用中，連続したクラスタの一部であり，後続するクラスタを持つ．値が，後続するクラスタの番号を示している．
5.25インチ両面倍密度のとき C1H～C8H 5.25インチ両面高密度と8インチ両面倍密度のとき C1H～DAH	使用中，連続したクラスタの最後のクラスタであり，下4(両面高密度または，8インチのときは下5)ビットの内容が，そのクラスタで実際に使われているセクタの数を表す．
FEH	予約済みのクラスタで，ファイルとして使うことはできない(ディスクコード，IPL，ディレクトリ，FAT自身を含むクラスタがこれである)．
FFH	未使用

5 ID

IDには1セクタ(＝256バイト)が割り当てられています．各バイトは右のような意味を持っています．

最初の1バイトは，そのフロッピーディスク全体の属性を示しています．値の意味は次のとおりです．

属性	ファイル数	BASICテキスト
1バイト	1バイト	254バイト

値(16進数)	意　味
0H	読み書き可
10H	書き込み禁止
40H	リードアフターライト

通常のシステムディスクでは，〈ファイル数〉として255(16進数ではFF)が指定されています．この場合，システム起動時にキーボードからファイル数を入力する必要があります．このファイル数に0～15を指定すると，BASICをスタートさせるときに

```
How many files(0-15)?
```

を表示せずに，自動的にそのファイル数が設定されます．

ファイル数の設定が終わると(ファイル数を設定せずに"How many files(0-15)?"に対してキーボードから入力を行った場合でも)，次のBASICテキストの内容が自動的に実行されます．つまり，BASICをスタートさせる場合，システムディスクのIDに書かれているBASICテキストの内容は常に実行されます．

IDの通常の状態は，ファイル数が255(FFH)，BASICテキストがすべて0か32(0Hか20H)になっています．

例

```
00 05 72 75 6E 22 41 00 ・・・・・・・・・・・・・・・・・・00
      (r)(u)(n)(")(A)(16進数)
```

この例のようにIDセクタの内容が書かれている場合，自動的に5個のバッファが用意され，"A"というプログラムが走り出します．

IDの書き換えを行いたい場合は，システムディスクに入っているディスクユーティリティを使ってください．IDの書き換え方法については，第16章1節をご覧ください．

6 PC-8801MHがさらに使用することのできるその他のフォーマットのフロッピーディスク

1節から5節まででは，次の3つのタイプのフロッピーディスクの構造に関して述べました．

5.25インチミニフロッピーディスク　(両面倍密度)*
5.25インチミニフロッピーディスク　(両面高密度)**
8インチフロッピーディスク　(両面倍密度)*

＊　一般に2Dと呼ばれています．
＊＊　一般に2HDと呼ばれています．

しかし，フロッピーディスクのフォーマットには，上の3つのほかにもいくつか種類があり，PC-8800シリーズやPC-9800シリーズ用の市販のソフトウェアや，BASIC DISK Versionなどで使われています．ここでは，PC-8801MHで両面倍密度のフロッピーディスクを使う方法と，さらに使用することのできる2つのタイプのフロッピーディスクについて説明をします．

5.25インチミニフロッピーディスク　(片面倍密度)***
5.25インチミニフロッピーディスク　(両面倍密度倍トラック)****

＊＊＊　一般に1Dと呼ばれています．
＊＊＊＊　一般に2DDと呼ばれています．

1．5.25インチミニフロッピーディスク(両面倍密度)

PC-8000シリーズ，PC-8800シリーズの従来機BASIC DISK Version，およびPC-8801FHのN₈₈-BASIC DISK Versionではこのフォーマットを用いています．

PC-8801MHでも，ディスクユーティリティの「システムディスクの作成」によって両面倍密度のN₈₈-BASICシステムディスクをつくり，これでN₈₈-BASIC DISK Versionをスタートすれば，従来機でつくったファイルを使用することができます．

5.25インチミニフロッピーディスク(両面倍密度)の構成と「システムディスクの作成」は，第16章の1節と7節を参照してください．

2．5.25インチミニフロッピーディスク(片面倍密度)

3. 5.25インチミニフロッピーディスク(両面倍密度倍トラック)

前ページの 2.は，PC-8801，PC-8001などに片面用ドライブPC-8031-1Wを接続したシステムで用いていたフォーマットです．3.は，PC-9800シリーズで使用しているフォーマットです．

これらのフォーマットで作られているファイルは，ディスクユーティリティの「ファイル転送」を使って，PC-8801MHで使用できるフォーマットに転送してから使います．

注意 上記のようにして，PC-8801MHで両面倍密度のファイルを読み込み／書き込みをすることができます．
ただし，このようにしてつくったファイルを従来機で読み込む場合，正常に動かないことがあります．

第16章

ディスク関係の
ユーティリティプログラム

N88-BASICシステムディスクには，N88-BASIC　DISK Version およびN88-日本語BASICでフロッピーディスクを扱うときに便利なユーティリティプログラムが入っています．

このディスクユーティリティは，システムディスクを作ったり，フロッピーディスクのコピーや，ファイル転送を行ったりするものです．

ディスクドライブが装備されていないPC-8801FHのモデル10をお持ちの方は，別売の増設ディスクドライブをお買い求めください．

1 ディスクユーティリティの概念

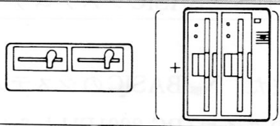
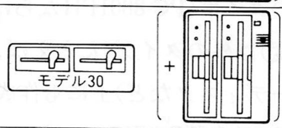
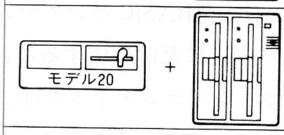
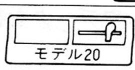
ディスクユーティリティは，以下の７つの働きをします．ここでは，それぞれの働きがどのようなものかを説明します．また，それぞれに必要なプログラムのファイル名を表にまとめておきます．

1．フロッピーディスクのコピー(その1)(copy2.j88/copy1.j88)

フロッピーディスクにセーブされたファイルは壊れることがあります．たとえば，間違えて必要なファイルをKILL命令で消してしまったり，DSKO$命令でファイルの内容を壊したり，また，ノイズなどハードウェアのトラブルなどで壊れる可能性もあります．

そこで，重要な情報が記録されているフロッピーディスクは，定期的にコピーを作っておいて，もしファイルが壊れても，そのコピーファイルを用いてその被害を最小限にとどめることが必要です．このようなコピーを行うために，このプログラムを使います．

なお，このプログラムはコピーを行う前に自動的に物理フォーマットを行います(実際の手順については，「ユーザーズガイド」の第3章を参照してください)．

	システム構成	もとになるディスク → コピーしてできるディスク	使用するプログラム
PC-8801MH		ミニ 2HD → ミニ 2HD 8インチ → 8インチ	
PC-8801FH		ミニ → ミニ 8インチ → 8インチ	dskut2.j88 dskut2＊bin
		8インチ → 8インチ	
		ミニ → ミニ	dskut1.j88 dskut1＊bin

2. フロッピーディスクのコピー(その2)(dskut2.j88/dskut1.j88)

このプログラムも1.と同じファイルのコピーを実行しますが，物理フォーマットを「行う」，「行わない」の選択ができます．すでに物理フォーマットが済んでいるフロッピーディスクであれば，いちいち物理フォーマットを行わないで，ファイルのコピーができるのです．したがって，このプログラムは物理フォーマットの済んでいるフロッピーディスクに対しては，1.のプログラムより，ファイルのコピーをする手間と時間が省けます．

3. システムディスクの作成(dskut2.j88/dskut1.j88)

N_{88}-BASIC DISK VersionやN_{88}-日本語BASICを使うときには，システムディスクが必要です．N_{88}-BASICシステムディスクは，ディスクコードを含んでいるディスクのことで，N_{88}-BASICのスタートのときに，このディスクコードが本体のメモリに読み込まれて，N_{88}-BASICがスタートします．

N_{88}-日本語BASICシステムディスクには，ディスクコードのほかに辞書や外字フォントなども書き込まれていて，日本語文字を入力する際にも使われます．

「システムディスクの作成」では，既存のシステムディスクから新しいフロッピーディスクに，IPLとディスクコードだけをコピーします(IPL，ディスクコードについては，第15章を参照してください)．N_{88}-日本語BASICシステムディスクの場合は，さらに辞書と外字フォントもコピーします．

このとき，もとになるシステムディスクがN_{88}-BASICのものなら，新しく作られるディスクも，N_{88}-BASICシステムディスクになります．N_{88}-日本語BASICシステムディスクを作る場合も同様の関係になります．

また，N_{88}-BASICのシステムディスクは，もとになるシステムディスクのタイプ(PC-8801FHならば両面倍密度，PC-8801MHならば両面高密度)以外のタイプ(たとえば，両面倍密度倍トラックや8インチフロッピーディスクなど)上にも作成することができます．

N_{88}-日本語BASICのシステムディスクは，同じタイプのフロッピーディスク上にだけ作成することができます．

なお，システムディスクは，N_{88}-BASICのスタートに使うだけでなく，プログラムやデータを書き込むこともできます．

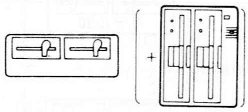
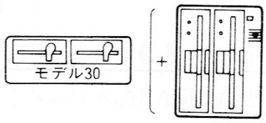
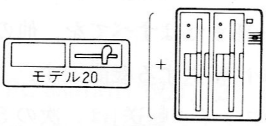
4. データディスクの作成(dskut2.j88/dskut1.j88)

買ったばかりのフロッピーディスクには，プログラムやデータを書き込むことはできません．未使用のフロッピーディスクにシステムフォーマットを行うと，読み書き可能になります．本機では「データディスクの作成」がこれにあたります．

これによって処理されたフロッピーディスクは，プログラムやデータの読み込み／書き込みができる状態になります．

ただし，データディスクはディスクコードを含んでいませんので，N₈₈-BASIC DISK VersionやN₈₈-日本語BASICを起動することはできません．

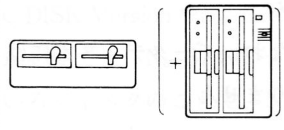
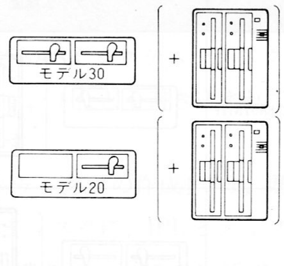
なお，データディスクには，N₈₈-BASICとN₈₈-日本語BASICの区別はありません．

	システム構成	データディスクにするフロッピーディスクのタイプ	必要なプログラム
PC-8801MH		ミニ 2HD 8インチ	
PC-8801FH		ミニ 8インチ	dskut2.j88 dskut2＊bin
			

5. IDセクタの書き換え (dskut2. j88/dskut1. j88)

システムディスクのIDセクタには，BASICのスタート時にBASIC側からたずねてくる"How many files(0-15)?"に対する値と，BASICがスタートした直後に自動的に実行させたいBASIC命令を書き込むことができます．

「IDセクタの書き換え」では，以下のシステムディスクのIDセクタの設定を行います．

	システム構成	IDセクタを書き換えるシステムディスク	必要なプログラム
PC-8801MH		ミニ 2HD N88 ミニ 2HD N88日本語 8インチ N88	dskut2.j88 dskut2＊bin
PC-8801FH		ミニ N88 ミニ N88日本語 8インチ N88	dskut2.j88 dskut2＊bin

6. ファイル転送 (xfiles. j88)

「ファイル転送」は，フロッピーディスクに書き込まれているファイルの一部またはすべてを，他の異なった種類のフロッピーディスクのファイルに転送する機能です．

「ファイル転送」は，次の５つのタイプ(ただし，PC-8801FHの場合，本体のみでは(1)と(2)のタイプは使えません)のフロッピーディスクの間で，どの方向にでもファイルをコピーします．

(1) 両面高密度ミニフロッピーディスク(2HD)

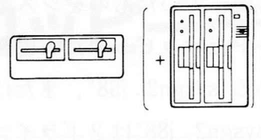
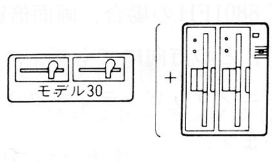
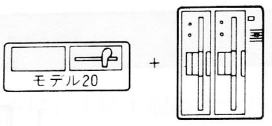
(2) 両面倍密度倍トラックミニフロッピーディスク(2DD)

(3) 両面倍密度ミニフロッピーディスク(2D)

(4) 片面倍密度ミニフロッピーディスク(1D)

(5) 8インチフロッピーディスク

これによって，本機と異なるタイプのフロッピーディスクを使用する機種で作ったファイルを使うことができます．

	システム構成	もとのファイルの書き込まれているフロッピーディスク	転送したファイルが書き込まれるフロッピーディスク	必要なプログラム
PC-8801MH		ミニ 注1 ミニ 注1 8インチ 8インチ	ミニ 注1 8インチ ミニ 注1 8インチ	xfiles . j88 xfiles＊bin
PC-8801FH		ミニ 注2 ミニ 注2 8インチ 8インチ	ミニ 注2 8インチ ミニ 注2 8インチ	
		8インチ ミニ 注2 8インチ	ミニ 注2 8インチ 8インチ	

注1）ミニフロッピーディスクは，2HD, 2DD, 2D, 1Dの４つのタイプが使用できます．

注2）ミニフロッピーディスクは，片面倍密度（1D），両面倍密度（2D）の２つのタイプが使用できます．

「ファイル転送」のもとのディスク，転送先のディスクには，N_{88}-BASICやN_{88}-日本語BASICなどのBASICの種類，システムディスク，データディスクといったフロッピーディスクの種類に制限はありません．

注意　PC-8801MHで(2)，(3)，(4)タイプのフロッピーディスクにファイルを書き込んだ場合，トラックの幅が異なるため，PC-8801MH以外の機種では正常に読み出せないことがあります．

"xfiles . j88"は，フロッピーディスクのフォーマットは行いませんから，ファイルが書き込まれる側のフロッピーディスクは，すでにフォーマットしてあるものでなければいけません．適当な，フォーマットされているフロッピーディスクがないときは，"dskut2 . j88"の「システムディスクの作成」，「データディスクの作成」を用いて，新しいフロッピーディスクをフォーマットしてご使用ください．

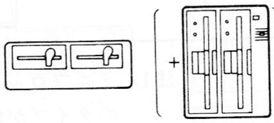
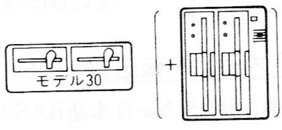
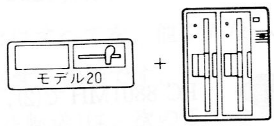
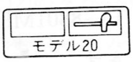
ファイルが書き込まれる側のフロッピーディスクは，すでにいくつかファイルが書き込まれているものでかまいません．"xfiles . j88"は，これらのファイルはそのままにして，さらに新しいファイルを追加していきます．ただし，転送先のフロッピーディスクに同じ名前のファイルがあったときは，新しいファイルに書き換えられてしまいますから注意してください．

7. ディスクコードのコピー(sysgn2.j88/sysgn1.j88)

「フロッピーディスクのコピー」や「ファイル転送」は，フロッピーディスクに入っているファイルをコピーします．しかし，データディスクとして使っていたものをシステムディスクにしたいときなど，ディスクコードだけコピーしたい場合には，N_{88}-BASICシステムディスクに入っている"sysgn2 . j88"，または"sysgn1 . j88"を使います．

"sysgn2 . j88"は2ドライブ用で，両面高密度ミニフロッピーディスク(PC-8801FHの場合，両面倍密度)および8インチフロッピーディスク間で，どの方向にでもディスクコードをコピーできます．

ミニフロッピーディスクドライブ1ドライブ用には"sysgn1 . j88"を使います．

	システム構成	もとになるシステムディスク → 新しく作られるシステムディスク	必要なプログラム
PC-8801MH		ミニ2HD N88 → ミニ2HD N88 ミニ2HD N88 → 8インチ N88 8インチ N88 → ミニ2HD N88 8インチ N88 → 8インチ N88	sysgn2 . j88 sysgn2 * bin
PC-8801FH	 モデル30	ミニ N88 → ミニ N88 ミニ N88 → 8インチ N88 8インチ N88 → ミニ N88 8インチ N88 → 8インチ N88	sysgn2 . j88 sysgn2 . bin
	 モデル20	ミニ N88 → 8インチ N88 8インチ N88 → ミニ N88 8インチ N88 → 8インチ N88	
	 モデル20	ミニ N88 → ミニ N88	sysgn1 . j88 sysgn1 * bin

2 フロッピーディスクの種類

本機に添付しているシステムディスクをそのまま使う場合，フロッピーディスクのタイプは，PC-8801MHならば5.25インチ両面高密度ミニフロッピーディスク(2HD)，PC-8801FHならば5.25インチ両面倍密度ミニフロッピーディスク(2D)になります．

また，本機に8インチフロッピーディスクユニットを接続することにより，さらに8インチ両面倍密度フロッピーディスク(2D)が使用できます．

本機のディスクユーティリティプログラムは，主に上記の3つのタイプのフロッピーディスクを対象にしますが，PC-8801FHの場合「システムディスクの作成」では，5.25インチ両面高密度ミニフロッピーディスク(2HD)をN_{88}-BASIC/N_{88}-日本語BASICシステムディスクにすることはできません．

あるいは「ファイル転送」では，上の3つのタイプのほかに，5.25インチ両面倍密度倍トラックミニフロッピーディスク(2DD)や5.25インチ片面倍密度ミニフロッピーディスク(1D)上のファイルを扱うことができます．異なる機種の間でBASICのプログラムファイルを転送する場合は，アスキーセーブしたものを使ってください．ただし，PC-8801FHでは2HDを使ってプログラムファイルを転送することはできません．

下図は，それぞれのミニフロッピーディスクの特徴などをまとめたものです(上記のフロッピーディスクとファイルの構造については，第15章をご覧ください)．

各フロッピーディスクのまとめ

ディスクタイプ	使用方法(BASIC)	使用機種	NEC製 注1 フロッピーディスク	読み書き中のドライブランプの色
5.25インチ両面高密度ミニフロッピーディスク(2HD)	添付のシステムディスクでBASICをスタートする．	● PC-8801MH ● 従来のPC-8800シリーズ＋PC-8831-MW ● PC-9800シリーズ	PC-9836-M	緑
5.25インチ両面倍密度倍トラックミニフロッピーディスク(2DD)	「ファイル転送」で，ファイルを2HDまたは2D上に移してから使う．	● PC-8801MH ● PC-9800シリーズ	PC-9836-4	赤
8インチ両面倍密度フロッピーディスク(2D)	8インチフロッピーディスクユニットを接続する．	● PC-8801MH ● PC-8801FH ● PC-8800シリーズ ● PC-9800シリーズ	PC-8886 (物理フォーマット済)	赤
5.25インチ両面倍密度ミニフロッピーディスク(2D)	注2)「システムディスクの作成」で2D用のシステムディスクを作り，これを使ってBASICをスタートする．	● PC-8801MH ● PC-8801FH ● PC-8800シリーズ ● PC-9800シリーズ ● PC-8000シリーズ	PC-8036-2	赤
5.25インチ片面倍密度ミニフロッピーディスク(1D)	「ファイル転送で，ファイルを2HDまたは2D上に移してから使う．	● PC-8801MH ● PC-8801FH ● PC-8800シリーズ＋PC-8031-1W ● PC-8000シリーズ＋PC-8031-1W	PC-8036 PC-8036-2	赤

注1) フロッピーディスクのタイプによって，それぞれ専用の製品が用意されています．タイプの違うものに使用した場合は，動作が保障されません．

注2) PC-8801FHの場合，添付のシステムディスクでBASICをスタートできます．

3 物理フォーマットとシステムフォーマット

1. 物理フォーマット

サーフェス，トラック，セクタ(第15章参照)といったアドレスをフロッピーディスクに付けることを「物理フォーマット」といいます．

未使用のフロッピーディスクは，必ず物理フォーマットを行わなければなりません．NECが供給している8インチフロッピーディスク(PC-8886)は物理フォーマット済みですので，その必要はありません．

物理フォーマットを行うと，DSKI$, DSKO$命令を使えるようになりますが，他のコマンド，ステートメントを使用することはできません．

2. システムフォーマット

FAT，ディレクトリ，IDの初期化を行うことを「システムフォーマット」といいます(FAT，ディレクトリ，IDについては本書第15章を参照してください)．システムフォーマットを行うことにより，そのフロッピーディスクに対して，BASICのコマンド，ステートメントを実行できるようになります．

5.25インチミニフロッピーディスクで買ってきたばかりの新しいものは，必ず上記の2段階のフォーマットをしないと，BASICで使うことはできません．

他のパソコンで使っていたフロッピーディスクを再利用する場合も，物理フォーマットが違っている場合がありますから，物理フォーマットからしなおした方がよいでしょう．この章で説明するディスクユーティリティのうち次のものは，必要に応じてフォーマットをしてくれます．

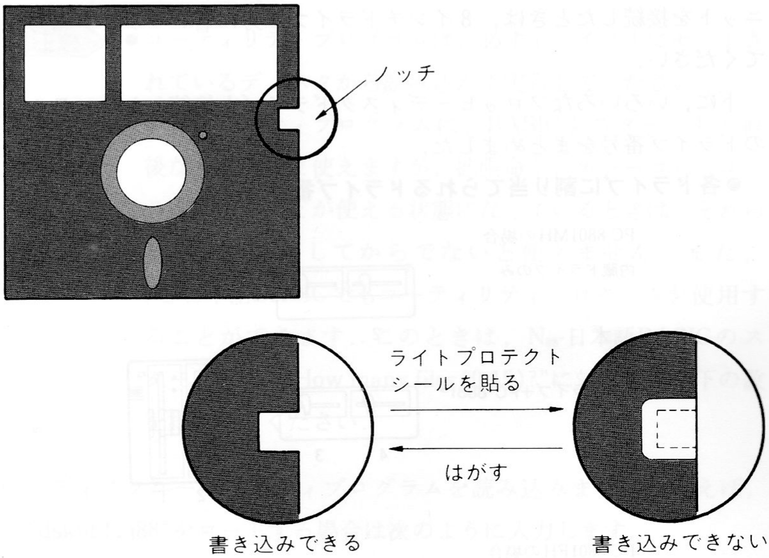
- 「フロッピーディスクのコピー」
- 「システムディスクの作成」
- 「データディスクの作成」

フォーマットを行うことは，フロッピーディスクを扱う上で大切なことですが，大切なデータが書き込まれたフロッピーディスクを誤ってフォーマットしてしまわないよう，特に注意しなければなりません．

フォーマットをすると，前に書き込まれたデータはすべて消されてしまいます．

4 フロッピーディスクのライトプロテクト(書き込み禁止)

ミニフロッピーディスクには，図のように切り込み(ノッチといいます)が付けられていて，この部分にライトプロテクトシールを貼ることにより，そのフロッピーディスクを書き込み禁止にする(これをライトプロテクトをかけるといいます)ことができます.



大切なシステムディスクや，重要なデータの書き込まれているフロッピーディスクには，誤操作で，もとのデータがなくなってしまわないように，ライトプロテクトシールを貼っておきましょう.

「フロッピーディスクのコピー」や「ディスクコードのコピー」をするときは，誤操作によってデータを消してしまわないように，もとになるフロッピーディスクには必ずライトプロテクトシールを貼っておくようにしましょう.

5 ドライブ番号

本体に内蔵されているドライブだけを使う場合は，向かって右側がドライブ番号1のドライブ，左側がドライブ番号2のドライブです．

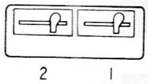
本機には，8インチフロッピーディスクユニットを接続することができます．8インチディスクユニットを接続したときは，8インチドライブの方に若いドライブ番号が割り当てられますから，注意してください．

下に，いろいろなフロッピーディスクドライブを接続したシステムを想定して，それぞれのドライブのドライブ番号をまとめました．

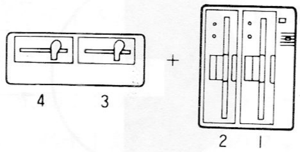
●各ドライブに割り当てられるドライブ番号

PC-8801MHの場合

内蔵ドライブのみ

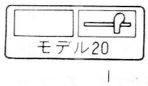


内蔵ドライブ＋PC-8881❶

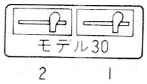


PC-8801FHの場合

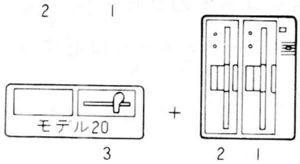
モデル20



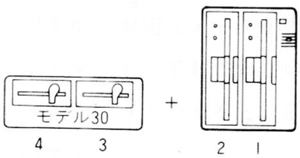
モデル30



モデル20＋PC-8881 ❶



モデル30＋PC-8881 ❶



❶ PC-8881と同じ機能で，ドライブが左右逆に付いているPC-8881(1)というモデルを接続することができます．このときは，左側がドライブ1，右側がドライブ2です．

上の図で，モデル20，モデル30とは，PC-8801FH本体にフロッピーディスクドライブがそれぞれ1台，2台内蔵されているものを意味します．つまり，モデル10に増設ドライブを1台付ければモデル20，2台付ければモデル30と同じになります．

6 ユーティリティプログラムのスタート

1. 8インチディスクユニットを接続していないとき

(1) N_{88}-BASICシステムディスクをドライブ1にセットして，N_{88}-BASIC DISK Versionをスタートさせます(V1モード，V2モードのどちらでもかまいません)．

注意
- ユーティリティプログラムは，必ずドライブ1にセットされているディスクから読み込んで実行してください．
- ユーティリティプログラムは，BASICをスタートした直後ならそのまま使えますが，拡張命令，カラーコピーユーティリティなどが使える状態になっているときは，それらをキャンセルしてからでないと使えません．また，N_{88}-日本語BASICでもユーティリティプログラムを使用することができます．このときは，N_{88}-日本語BASICのスタート時に，"How many files(0-15)?"に対して6以下の数を指定してください．

(2) ディスクユーティリティプログラムを読み込みます．たとえば，"dskut2 . j88"をロードする場合は次のように入力します．

```
load "dskut2 . j88"
```

次に，run と入力してプログラムを実行させます．

次のようなメッセージを表示しながら，必要な機械語プログラムを読み込みます(ここでは，ファイル名に"bin"という拡張子が付いている機械語のプログラムを読み込んでいます)．

```
機械語プログラムを読み込み中です。しばらくお待ちください。
```

読み込みが終わると，ユーティリティプログラムが始まります(もし，ドライブ1に機械語プログラムが入っていなかったら「ファイル"dskut2＊bin"が見つかりません。」というメッセージが表示されます．このようなときは，もう1度，機械語プログラムの入っているシステムディスクを使ってスタートをやりなおしてください)．

2. 8インチ ディスクユニットを接続しているとき

(1) N_{88}-BASICシステムディスクを使って，N_{88}-BASIC DISK Versionをスタートさせます．

注意
- ユーティリティプログラムは，BASICをスタートした直後ならそのまま使えますが，拡張命令，カラーコピーユーティリティなどが使える状態になっているときは，それらをキャンセルしてからでないと使えません．
- N_{88}-日本語BASICでもユーティリティプログラムを使用することができます．このときは，N_{88}-日本語BASICのスタート時に"How many files(0-15)?"に対して6以下の数を指定してください．

初めて8インチディスクユニットを使うときは，ミニフロッピーディスクドライブにシステムディスクをセットしてBASICをスタートします．

(2) ディスクユーティリティプログラムを読み込みます．たとえば"dskut2.j88"をドライブ3から読み込む場合は，次のように入力します．

load "3:dskut2.j88" ⏎

(3) 次に，run ⏎ と入力してプログラムを実行します．

(4) 画面に次のメッセージが現れます．

```
ユーティリティプログラムのはいっているドライブ番号？ ■
```

ドライブ3に入っている場合は3 ⏎ と入力します．

(5) 8インチディスクユニットを接続していないときと同様に，次のようなメッセージを表示しながら，必要な機械語プログラムを読み込みます．

```
機械語プログラムを読み込み中です。しばらくお待ちください。
```

(6) 読み込みが終わると，ユーティリティプログラムが始まります．

■ ユーティリティプログラムを終了したあと，もう一度スタートする

一度終了の操作を行ったあとでも，ユーティリティプログラムを読み込んだ状態ならば，RUN コマンドで何度でも再スタートすることができます．ただし，ミニフロッピーディスクドライブだけのシステムならドライブ1から，また，8インチディスクユニットが接続されているシステムなら指定したドライブから機械語プログラムを読み込んでプログラムをスタートさせますから，システムディスクのセットを忘れないようにしてください．

7 各ユーティリティの操作手順

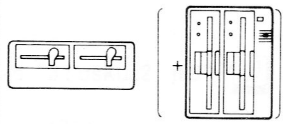
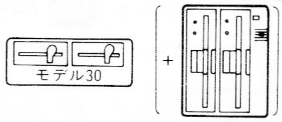
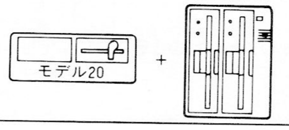
各ユーティリティの使い方を説明します．システムの構成によって，使用するプログラムが違いますから注意してください．

もし誤った操作をしてしまったら，［停止］を押して，もう1度プログラムを実行してください．

1. フロッピーディスクのコピー

(1) dskut2．j88

ここでは，次表のようなシステム構成の場合のフロッピーディスクのコピーを説明します．

	システム構成	もとになるディスク		コピーしてできるディスク	使用するプログラム
PC-8801MH		ミニ 2HD	➡	ミニ 2HD	dskut1．j88 dskut1＊bin
		ミニ 2D	➡	ミニ 2D	
		8インチ	➡	8インチ	
PC-8801FH	 	ミニ 2D	➡	ミニ 2D	
		8インチ	➡	8インチ	

まず，"dskut2．j88"をスタートします．

画面には次のようにメニューが表示されます．

```
＊ ＊　ディスク　ユーティリティ　＊ ＊

　1　フロッピーディスクのコピー
　2　システムディスクの作成
　3　データディスクの作成
　4　IDセクタの書き換え
　5　終了

実行したい処理の番号を入力してください？ ■
```

キーボードから1 ↵ と入力してください．すると，画面には次のように表示されます．

```
【　フロッピーディスクのコピー　】
コピーしたいフロッピーディスクをセットしたドライブ番号？■
```

コピーしたいフロッピーディスク(たとえば，N88-BASICシステムディスクなど)をドライブにセットし，そのドライブ番号を入力します．ドライブ1にセットした場合は，1 ↵ と入力します(ドライブ番号の指定に誤りがあるときは，エラーメッセージが表示されて，直前の画面に戻ります)．

次に，新しいディスクをドライブにセットするように指示してきます．

```
新しいフロッピーディスクをセットしたドライブ番号？■
```

新しいフロッピーディスク(または，すべて消してもかまわないフロッピーディスク)をセットし，そのドライブ番号を入力します．たとえばドライブ2の場合は，2 ↵ と入力します．

次に物理フォーマットをするかどうか聞いてきます．

```
物理フォーマットをしますか（y／n）？■
```

物理フォーマットをする必要がある場合(特に，開封したばかりの新しいミニフロッピーディスク，または別のことに使用していたミニフロッピーディスク)には，ここで

　　y ↵

と入力します．n ↵ と入力すると，物理フォーマットをせずにコピーが始まります(ユーティリティプログラムの場合，y ↵ は，yを省略して ↵ だけでもかまいません)．

物理フォーマットをすると，前に書き込まれていたデータはすべて消えてしまうので，ここでもう1度確認のため，次のメッセージが表示されます．

```
ドライブ2のディスクをフォーマットします。
よろしいですか（y／n）？ ■
```

y ⏎ と入力すれば，物理フォーマットが始まります．また，n ⏎ と入力すると，直前の画面に戻ります．

```
物理フォーマット中です。
```

物理フォーマットが終わるとコピーが始まります．

コピーされるフロッピーディスクの入っているドライブと，新しいフロッピーディスクの入っているドライブランプが順に点灯すると同時に，画面にコピーしているトラック番号が表示されます．たとえばドライブ1からドライブ2にコピーする場合なら，次のようになります．

```
コピー　　ドライブ　1　→　ドライブ　2
トラック　0
トラック　1
トラック　2
　　　　　・
　　　　　・
　　　　　・
```

注意
- トラック番号は連続していないこともあります．
- 8インチフロッピーディスクをフォーマットするときは，トラック番号を表示しながら行います．
- フロッピーディスクが汚れていたり，傷が付いたりしている場合は，"I/Oエラー"と表示されてコピーが中断します．他のフロッピーディスクを使ってコピーしなおしてください．

無事に終了すれば次のように表示されます．

```
終了しました。
ディスクユーティリティの作業を続けますか（y／n）？ ■
```

y ⏎

と入力すれば，"dskut2．j88"のメニュー画面に戻り，

n ⏎

と入力すれば，"dskut2．j88"を終了します．

(2) dskut1．j88

ここでは，ミニフロッピーディスクドライブ1台だけを使ってフロッピーディスクのコピーを行う手順を説明します．

	システム構成	もとになるディスク		コピーしてできるディスク	使用するプログラム
PC-8801FH	モデル20	ミニ 2D	➡	ミニ 2D	dskut1.j88 dskut1＊bin

まず，"dskut1．j88"をスタートします．

画面には次のようにメニューが表示されます．

```
＊ ＊　 ディスク　ユーティリティ　 ＊ ＊　 （1ドライブ用）

　　　1　フロッピーディスクのコピー
　　　2　システムディスクの作成
　　　3　終了

　　　実行したい処理の番号を入力してください？ ■
```

キーボードから1⏎と入力します．

画面には次のように表示されます．

```
【　フロッピーディスクのコピー　】
物理フォーマットをしますか（y／n）？ ■
```

新しいミニフロッピーディスクの物理フォーマットが必要な場合は，

y⏎

と入力します（n⏎と入力すると，物理フォーマットをせずにコピーが始まります）．

物理フォーマットを行う場合は，

```
新しいフロッピーディスクをドライブにセットしてください。
よろしいですか（y／n）？ ■
```

と表示されますので，新しいミニフロッピーディスクをドライブにセットし，

y⏎

と入力してください（n⏎と入力すると，"dskut1．j88"プログラムが終了します）．

```
物理フォーマットを行っています。
```

と表示されて，物理フォーマットが行われます．

次に，すべてのファイルがコピーされるまで，以下の①と②の操作を繰り返します．

① 次のように表示されたら，コピーされるフロッピーディスクを入れます．

```
マスターディスクをドライブにセットしてください。
よろしいですか（y／n）？ ■
```

正しくセットできたら，

y ⏎

を入力してください（n ⏎ と入力すると，"dskut1．j88"プログラムが終了します）．

```
読み込みを行っています。
トラック        0
トラック        1
トラック        2
                ・
                ・
                ・
```

と表示されながら，データが読み込まれます．

② 次に，

```
読み込みが終わりました。

新しいフロッピーディスクをドライブにセットしてください。
よろしいですか（y／n）？ ■
```

と表示されたら，マスターディスクを取り出して，新しいミニフロッピーディスクを入れます．

正しくセットできたら，

y ⏎

を入力してください（n ⏎ と入力すると，"dskut1．j88"プログラムが終了します）．

```
書き込みを行っています。
トラック        0
トラック        1
トラック        2
                ・
                ・
                ・
```

と表示されながらデータが書き込まれます．書き込みが終了すると

書き込みが終わりました。

と表示されます．①の操作に戻ります．

③　すべてのファイルがコピーされると，

```
終了しました。
ディスクユーティリティの作業を続けますか（y／n）？ ■
```

と表示され，終了します．

y 

と入力すれば "dskut1．j88" のメニュー画面に戻り，

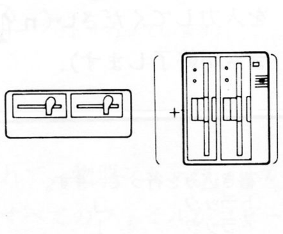
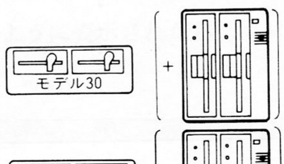
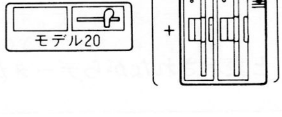
n 

と入力すれば "dskut1．j88" を終了します．

2. システムディスクの作成

(1)　dskut2．j88

ここでは，次表のようなシステム構成の場合のシステムディスクの作成の手順を説明します．

	システム構成	もとになるシステムディスク		新しく作られるシステムディスク		必要なプログラム
PC-8801MH		ミニ2HD	N88	ミニ2HD	N88	dskut2.j88 dskut2＊bin
		ミニ2HD	N88	ミニ2D	N88	
		ミニ2HD	N88 日本語	ミニ2HD	N88 日本語	
		ミニ2HD	N88	8インチ	N88	
		8インチ	N88	ミニ2HD	N88	
		8インチ	N88	ミニ2D	N88	
		8インチ	N88	8インチ	N88	
PC-8801FH		ミニ2D	N88	ミニ2D	N88	
		ミニ2D	N88 日本語	ミニ2D	N88 日本語	
		ミニ2D	N88	8インチ	N88	
		8インチ	N88	ミニ2D	N88	
		8インチ	N88	8インチ	N88	

まず，"dskut2 . j88"をスタートします．

画面には次のようにメニューが表示されますから，

```
＊　＊　ディスク　ユーティリティ　　＊　＊

1　フロッピーディスクのコピー
2　システムディスクの作成
3　データディスクの作成
4　IDセクタの書き換え
5　終了

実行したい処理の番号を入力してください？ ■
```

2 ⏎ と入力します．すると，画面には次のように表示されます．

```
【　システムディスクの作成　】
システムディスクをセットしたドライブ番号？　■
```

もとになるシステムディスクをドライブにセットし，そのドライブ番号を入力します．ここでセットしたシステムディスクと同じタイプのシステムディスクが作られます（N_{88}-BASIC → N_{88}-BASIC, N_{88}-日本語BASIC → N_{88}-日本語BASIC）．

たとえば，ドライブ１の場合は1 ⏎ と入力します．

すると，画面に

```
新しいフロッピーディスクをセットしたドライブ番号？ ■
```

と表示されますので，新しいフロッピーディスクをドライブにセットし，そのドライブ番号を入力します．たとえば，ドライブ２の場合は2 ⏎ と入力します．

注意 N₈₈-BASICでは，システムディスクと新しいフロッピーディスクは，ミニ，8インチのように種類が違っていてもかまいません．

N₈₈-BASICシステムディスクを作成するときのみ，どちらのタイプのシステムディスクを作るかを指定します．もとになるシステムディスクと異なるタイプでもかまいません．

注意 PC-8801FHの場合，システムディスクのタイプの指定はできません．

```
作成するシステムディスクのディスクタイプを指定してください。
    1.  5.25インチ両面高密度(2HD)
    2.  5.25インチ両面倍密度(2D)

ドライブ2のディスクタイプ? ■
```

たとえば，両面高密度(2HD)のシステムディスクにしたいなら **1** ⏎ と入力します(PC-8801FHでは，この画面は表示されません)．

システムディスクのタイプが，これでよいかどうかを確認してきます．

```
5.25インチ両面高密度(2HD)タイプのシステムディスクを作成します。
よろしいですか(y/n)? ■
```

これでよければ，**y** ⏎ と入力します(**n** ⏎ と入力すると，直前の画面に戻ります)．

次に，物理フォーマットをするかどうか聞いてきます．

物理フォーマットの必要がある場合は，**y** ⏎ と入力します(**n** ⏎ と入力すると，システムフォーマットに移ります)．

```
物理フォーマットをしますか(y/n)? ■
```

確認のためにもう1度メッセージが表示されます．

```
ドライブ2のディスクをフォーマットします。
よろしいですか（y／n）？ ■
```

フォーマットしてよければ，

y 

と入力します(物理フォーマットをすると，前に書き込まれていたデータはすべて消えてしまうため，ここでもう1度確認しています．n  と入力すると，直前の画面に戻ります)．

次のようなメッセージが表示されて，物理フォーマットが始まります．

```
物理フォーマット中です。
```

物理フォーマットが終わると，続けて次のようなメッセージが表示されてシステムフォーマットが行われ，FAT，ディレクトリ，IDが初期化されます．

```
システムフォーマット中です。
```

システムフォーマットが終わると，IPLとディスクコード(システム)のコピーをします．

```
システムをコピー中です。
```

と表示されて，システムのコピーが行われます．

次のようにして表示されれば終わりです．

```
終了しました。
ディスクユーティリティの作業を続けますか（y／n）？ ■
```

y ↵

と入力すれば，"dskut2．j88"のメニュー画面に戻り，

n ↵

と入力すれば，"dskut2．j88"を終了します．

(2) dskut1．j88

ミニフロッピーディスクドライブを1台だけ使ってシステムディスクを作成します．

もとになるシステムディスクと，新しいフロッピーディスクを1つのドライブに交互にセットしながら操作します．

	システム構成	もとになる システムディスク → 新しく作られる システムディスク	必要な プログラム
PC-8801FH	モデル20	ミニ N88 → ミニ N88 ミニ N88日本語 → ミニ N88日本語	dskut1.j88 dskut1＊bin

まず，"dskut1．j88"をスタートします．

画面には次のように表示されます．

```
＊ ＊   ディスク  ユーティリティ   ＊ ＊   （1ドライブ用）

        1  フロッピーディスクのコピー
        2  システムディスクの作成
        3  終了

        実行したい処理の番号を入力してください?
```

2 と入力すると，次のメッセージが現れます．

```
【　システムディスクの作成　】
システムディスクをドライブにセットしてください。
よろしいですか（y／n）？ ■
```

ここで，もとになるシステムディスクをドライブにセットし，

y

と入力します（n を入力すると "dskutl . j88" プログラムが終了します）．すると，

```
新しいフロッピーディスクをドライブにセットしてください。
よろしいですか（y／n）？ ■
```

と表示されますので，新しいミニフロッピーディスクをドライブにセットし，y と入力してください．

次に，

```
物理フォーマットをしますか（y／n）？ ■
```

と表示されますので，新しいミニフロッピーディスクの物理フォーマットが必要な場合は，

y

と入力します（n と入力すると，物理フォーマットをせずにコピーを始めます）．

```
物理フォーマットを行っています。
```

と表示されて，物理フォーマットが行われます．

次に，

```
システムフォーマットを行っています。
```

と表示され，システムフォーマットが行われます．

```
システムディスクをドライブにセットしてください。
よろしいですか（y／n）？ ■
```

と表示されますので，システムディスクをセットし，

y ⏎

と入力します．

```
ディスクコードを読み込んでいます。
```

と表示され，ディスクコードが読み込まれます．

ディスクコードが読み込まれると，次のように表示されます．

```
読み込みが終わりました。

新しいフロッピーディスクをドライブにセットしてください。
よろしいですか（y／n）？ ■
```

ここでシステムディスクを取り出し，物理フォーマットが終わったミニフロッピーディスクをセットします．正しくセットしたら，**y** ⏎と入力します．

```
ディスクコードを書き込んでいます。
```

と表示され，ディスクコードが書き込まれます．

ディスクコードが書き込まれると，次のように表示されます．

```
書き込みが終わりました。
```

すべてのディスクコードが書き込まれると，次のように表示されます．

```
終了しました。
ディスクユーティリティの作業を続けますか（y／n）？ ■
```

y ⏎ と入力すれば，"dskut1．j88"のメニュー画面に戻り，n ⏎ と入力すれば，"dskut1．j88"を終了します．

3. データディスクの作成

どのようなフロッピーディスクドライブを持つシステムでも，"dskut2．j88"を使ってデータディスクを作ることができます．

まず，"dskut2．j88"をスタートします．

画面には次のようにメニューが表示されます．

```
＊ ＊　ディスク ユーティリティ　＊ ＊

　1　フロッピーディスクのコピー
　2　システムディスクの作成
　3　データディスクの作成
　4　ＩＤセクタの書き換え
　5　終了
　実行したい処理の番号を入力してください？ ■
```

3 ⏎ と入力すると，次のメッセージが現れます．

新しいフロッピーディスクをドライブにセットし，そのドライブ番号

を入力します.

```
【 データディスクの作成 】
新しいフロッピーディスクをセットしたドライブ番号? ■
```

次に物理フォーマットをするかどうか聞いてきます.

```
物理フォーマットをしますか(y/n)? ■
```

物理フォーマットが必要な場合は,

y ⏎

と入力します.

y ⏎ と入力すると,たとえば新しいフロッピーディスクがドライブ2にセットされていれば

```
ドライブ2のディスクをフォーマットします。
よろしいですか(y/n)? ■
```

と確認してきます.実行する場合は,y ⏎ と入力してください.

次のメッセージが表示されて,物理フォーマットが始まります.

```
物理フォーマット中です。
```

次にシステムフォーマットが行われます.

```
システムフォーマット中です。
```

データディスクを作り終わると，次のように表示されます．

```
終了しました。
ディスクユーティリティの作業を続けますか（y／n）？ ■
```

y ⏎

と入力すれば，"dskut2 . j88"のメニュー画面に戻り，

n ⏎

と入力すれば，"dskut2 . j88"を終了します．

4. IDセクタの書き換え

まず，"dskut2 . j88"をスタートします．

画面には次のように表示されますから，

```
＊＊　ディスク　ユーティリティ　＊＊

1　フロッピーディスクのコピー
2　システムディスクの作成
3　データディスクの作成
4　IDセクタの書き換え
5　終了

実行したい処理の番号を入力してください？ ■
```

4 ⏎ と入力します．

画面に次のメッセージが現れます．

```
【　IDセクタの書き換え　】
システムディスクをセットしたドライブ番号？ ■
```

IDセクタを書き換えたいシステムディスクをドライブにセットし，そのドライブ番号を入力します．たとえばドライブ2にセットした場合なら，2 [↵] と入力します．

次に，IDセクタに書き込む内容を設定していきます．

```
同時にオープンする最大ファイル数（0－15，設定しないときは－1）？ ■
```

とたずねてくるので，BASICのスタート時に"How many files(0-15)?"というプロンプトに対して答え，同時にオープンしたい最大のファイル数を入力します（[↵]だけを入力すると，0がセットされたことになります）．

たとえば，3つオープンさせたい場合なら，

3 [↵]

と入力します．

このとき－1を入力すると，IDのファイル数には255(16進でFFH)がセットされます．この場合，ファイル数はセットされず，BASICをスタートするたびにファイル数を尋ねるようになります．

次に

```
実行したいＢＡＳＩＣ命令を入力してください。
```

と表示されますから，BASICのスタートと同時に実行したいBASIC命令(コマンドやステートメント)を入力します(半角文字だけが入力可能です．ここでは，N_{88}-日本語BASICで使われる文字を入力することはできません)．

たとえば，"test . n88"というプログラムをスタートさせたい場合は

run "test . n88" [↵]

と入力します．

もし，ここで入力したBASICの命令が長すぎた場合は

```
限度（２３６バイト）を越えています。
```

と表示してブザーを鳴らし，もう1度BASIC命令を入力するように指示してきます．

以上の入力が終わると，確認のために，今までセットした内容が表示されます(オープンするファイル数を−1と入力した場合，"オープンするファイル数はスタート時に入力。"と表示されます)．

```
【　IDセクタの書き換え　】

オープンするファイル数.............

実行するBASIC命令.............



よろしいですか（y／n）？ ■
```

この内容でIDセクタを書き込んでよければ，

y

と入力します(やりなおしたいときは，n と入力すれば，直前の画面に戻ります)．すると，次のように表示され，セットした内容が書き込まれます．

```
書き換えを行っています。
```

IDセクタの書き換えが終了し，次のように表示されます．

```
終了しました。
ディスクユーティリティの作業を続けますか（y／n）？ ■
```

y

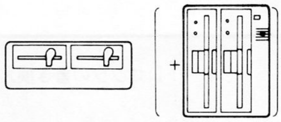
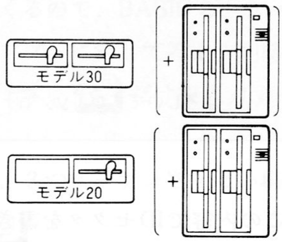
と入力すれば，"dskut2 . j88"のメニュー画面に戻り，

n

と入力すれば，"dskut2 . j88"を終了します．

5. ファイル転送

ここでは，次表のようなファイルの一部またはすべてをコピーする手順を説明します．

	システム構成	もとのファイルの書き込まれているフロッピーディスク		転送したファイルが書き込まれるフロッピーディスク	必要なプログラム
PC-8801MH		ミニ	➡	ミニ	xfiles. j88 xfiles＊bin
		ミニ	➡	8インチ	
		8インチ	➡	ミニ	
		8インチ	➡	8インチ	
		フロッピーディスクのタイプは2HD, 2D, 2DD, 1Dの4つが使えます．			
PC-8801FH		ミニ	➡	ミニ	
		ミニ	➡	8インチ	
		8インチ	➡	ミニ	
		8インチ	➡	8インチ	
		フロッピーディスクのタイプは2D, 1Dの2つが使えます．			

まず，"xfiles . j88"をスタートします．
画面には次のように表示されます．

```
<<  ファイル  転送  >>
マスターディスクのドライブ番号？ ■
```

転送されるもとのファイルが書き込まれているディスク(マスターディスク)をドライブにセットし，そのドライブ番号を入力します．

注意 フロッピーディスクは，それぞれシステムフォーマットされているディスクでなければなりません．

たとえばドライブ1なら，**1** ⏎ と入力します．

```
転送先ディスクのドライブ番号？ ■
```

転送されたファイルが書き込まれるフロッピーディスクをセットして，そのドライブ番号を入力します．たとえばドライブ2なら**2** ⏎ と入力します(この場合，すでにファイルが書き込まれていても，それらは，そのままで新しいファイルを追加していきます)．

次に，ミニフロッピーディスクを使っている場合は，ディスクのタイプを聞いてきます．

```
5．25インチフロッピーディスクのディスクタイプを指定してください。
      1．  両面高密度（2HD）
      2．  両面倍密度倍トラック（2DD）
      3．  両面倍密度（2D）
      4．  片面倍密度（1D）

ドライブ1のディスクタイプ？  2
ドライブ2のディスクタイプ？  ■
```

マスターディスク，転送先ディスク，それぞれのタイプを1～4で答えます(PC-8801FHの場合，1. 両面倍密度(2D)，2. 片面倍密度(1D)の2種類のみが表示されます)．

ここでは，ドライブ1に2DDタイプのフロッピーディスクを，ドライブ2に2HDタイプを入れたとします．

次のように確認のメッセージが現れます．

```
マスターディスク・・・・ドライブ1  （5．25インチ両面倍密度倍トラック）
転送先ディスク・・・・・ドライブ2  （5．25インチ両面高密度）

ドライブ1からドライブ2へファイルを転送します。
よろしいですか（y／n）？ ■
```

これでよければ，y ↵ と入力します（指定に誤りがあれば，n ↵ と入力してファイルの転送の作業を初めからやりなおします）．

次のように聞いてきます．

```
すべてのファイルを転送しますか（y／n）？ ■
```

マスターディスクの中の一部のファイルを転送するのなら，

n ↵

と入力します（ファイルをすべて転送する場合は y ↵ と入力します）．

転送するファイルを選びます．

```
<< ファイル 転送 >>
    File01.bas   <－■
    File02.bas
    File03.bas
    File04.bas
    File05.bas
    File06.bas
    File07.bas

ファイルの選択：上下カーソル ＋ リターンキー
    実行      ：スペースキー
```

ファイル名が枠の中に7つまで表示されますから，↑ と ↓ を使って，転送したいファイル名の位置まで矢印を移動し，↵ を押します．ファイル名はリバース表示になり，選択されます．複数のファイル名を続けて選択できます．一番下まで矢印を移動して，さらに ↓ を押すと，枠内のファイル名がスクロールしていきます（ファイル名にグラフィックシンボルを使っている場合，漢字やひらがななどの全角文字に変換されて表示される場合があります）．

1度選択されたファイルの転送をやめたいときは，もう1度矢印を合わせて ↵ を押します．

転送したいファイルを選び終わったら，［スペース］を押します．

確認を求めるメッセージが現れます．

```
よろしいですか（y／n）？■
```

これでよければ，

y ⏎

と入力します（n ⏎ と押すと，直前の画面に戻ります）．

ファイル名を表示しながら転送が行われます．

注意　もし，転送されるフロッピーディスクの未使用セクタ数より，マスターディスクが使っているセクタ数の方が大きい場合は，途中で"ディスクがいっぱいです。"というエラーメッセージが表示されます．このプログラムを実行する前に，転送できるか確認するようにしてください．FILES文によって各ファイルの大きさ（クラスタ数）を，DSKF関数によってフロッピーディスクの空き領域の大きさ（クラスタ数）を調べることができます．また，転送されるフロッピーディスクに同じファイル名のものがあると，そのファイルは転送するファイルの内容に置き換わりますから，注意してください．

```
<< ファイル 転送 >>
ドライブ1からドライブ2へ転送中です。

        File02.bas        1  →  2
        File04.bas        1  →  2
        File05.bas        1  →  2
            ⋮
```

転送が終わると，終了のメッセージが表示されます．

```
終了しました。
ファイル転送ユーティリティの作業を続けますか（y／n）？ ■
```

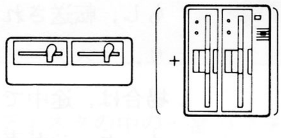
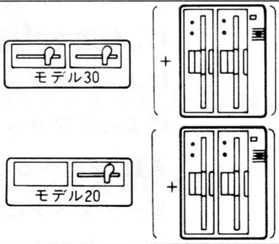
n ⏎

と入力すれば，"xfiles.j88"プログラムが終了します（続けてファイル転送を行いたいときはy ⏎ と入力します）．

6. ディスクコードのコピー

(1)　sysgn2 . j88

ここでは，次表のようなシステム構成の場合でのディスクコードのコピー手順を説明します．

	システム構成	もとになるシステムディスク → 新しく作られるシステムディスク	必要なプログラム
PC-8801MH		ミニ2HD N88 → ミニ2HD N88 ミニ2HD N88 → 8インチ N88 8インチ N88 → ミニ2HD N88 8インチ N88 → 8インチ N88	sysgn. j88 sysgn＊bin
PC-8801FH		ミニ2D N88 → ミニ2D N88 ミニ2D N88 → 8インチ N88 8インチ N88 → ミニ2D N88 8インチ N88 → 8インチ N88	

まず，"sysgn2 . j88"をスタートします．

画面には次のように表示されます．

```
<< ディスクコードのコピー >>
システムディスクをセットしたドライブ番号？ ■
```

もとになるシステムディスクをセットし，そのドライブ番号を入力します．たとえばドライブ1なら，**1** ↵ と入力します．

```
コピー先のフロッピーディスクをセットしたドライブ番号？ ■
```

メッセージに従って，ディスクコードを書き込みたいフロッピーディスク(すでにフォーマットされているフロッピーディスクでなければなりません．データディスクとして使っているものや，バージョンの違うN88-BASICシステムディスクなどが対象になります．コピー先のディスクに入っているファイルはそのまま残ります)をセットして，そのドライブ番号を入力します．

たとえばドライブ2なら，

2 ↵

と入力します．

確認のメッセージが表示されます．

```
コピー　ドライブ　1　→　ドライブ　2
よろしいですか（y／n）？ █
```

ディスクコードのコピーを始めてよければ，

y ⏎

と入力します（**n** ⏎ と入力すると直前の画面に戻ります）．

```
ただいまコピー中です。
```

と表示されてディスクコードのコピーが行われます．ただし，コピー先のディスク上にディスクコードを書き込む領域が十分にない場合などは，エラーメッセージが表示され，ディスクコードのコピーは中断します．

コピーが終了すると次のように表示されます．

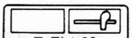
```
コピーが終了しました。
ディスクコードのコピー作業を続けますか（y／n）？ █
```

n ⏎

と入力すると，"sysgn2．j88"が終了します．ほかにもディスクコードをコピーしたいディスクがあれば，**y** ⏎ と入力してください．

(2)　sysgn1．j88

ミニフロッピーディスクドライブ１台だけを使って，ディスクコードのコピーをする手順を説明します．

	システム構成	もとになるシステムディスク　新しく作られるシステムディスク	必要なプログラム
PC-8801FH	 モデル20	ミニ 2D N88 ➡ ミニ 2D N88	sysgn1. j88 sysgn1＊bin

まず，"sysgn1．j88"をスタートします．

画面には次のように表示されます．

```
<< ディスクコードのコピー >> （1ドライブ用）

システムをコピーします。
システムディスクをドライブにセットしてください。
よろしいですか（y／n）？ ■
```

ドライブに，もとになるシステムディスクをセットし，

y

と入力します（n と入力すると，"sysgn1．j88"プログラムは終了します）．

```
システムの読み込みを行っています。
```

と表示して読み込みを行います．終了すると次のメッセージが現れます．

```
読み込みが終了しました。

書き込みを行いたいフロッピーディスクをドライブにセットしてください。
よろしいですか（y／n）？ ■
```

ディスクコードを書き込みたいフロッピーディスクをドライブにセットして，

y

と入力します．

```
システムの書き込みを行っています。
```

と表示して，書き込みが行われます．

ディスクコードのコピーが終了すると，次のメッセージが現れます．

```
書き込みが終了しました。

すべて終了しました。
ディスクコードのコピー作業を続けますか（y／n）？ ■
```

n ⏎

と入力すると"sysgnl．j88"プログラムが終了します（ほかにもディスクコードをコピーしたいディスクがあれば，y ⏎ と入力します）．

第17章

コンピュータコミュニケーション

本機は，背面にRS-232Cインタフェースと呼ばれるコネクタを持っています．RS-232Cインタフェースは，RS-232Cケーブル（クロス）で直接コンピュータ同士をつないだり，音響カプラやモデムを接続した上で電話回線や専用回線を介し，遠く離れたコンピュータとプログラムやデータのやりとりをするためのものです．

このRS-232Cインタフェースを使って，他のコンピュータとコミュニケーションを行う方法として，次の2つがあります．

① BASICモードにより，RS-232Cを1つのファイルとして扱い，BASICの管理下において使用する．

② ターミナルモードにより，外部のコンピュータのターミナルとして使用する．

本章では，上記の順に従って，通信手続きをご紹介します．

1 BASICモードでの通信

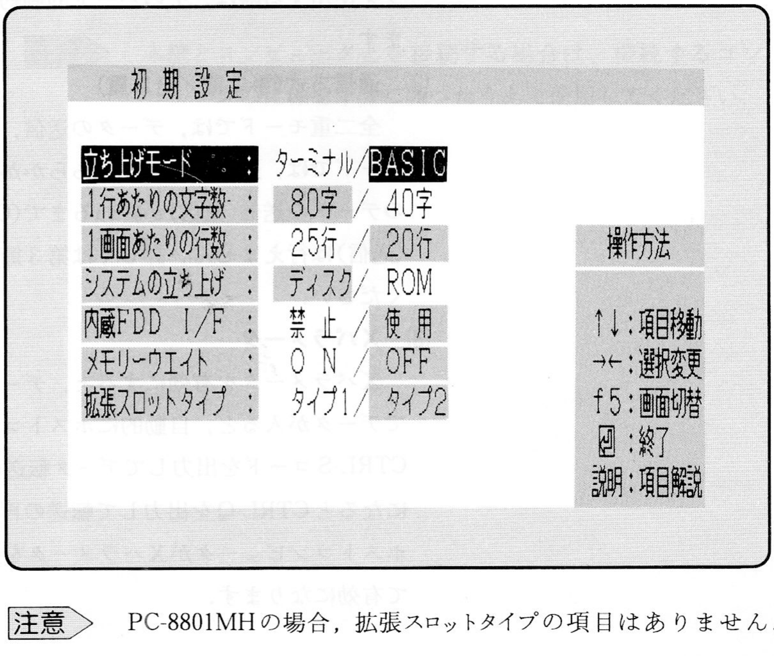
ここでは，BASICモードによるRS-232Cを1つのファイルとして扱い，BASICの管理下において使用する方法を説明します．

RS-232Cを使ってファイル操作をする場合も，基本的にはシーケンシャルファイルと同様に扱えます．つまり，一般の入出力命令が使用できるわけですが，他に細かい注意がいろいろと必要になってきます．

1．通信仕様の設定

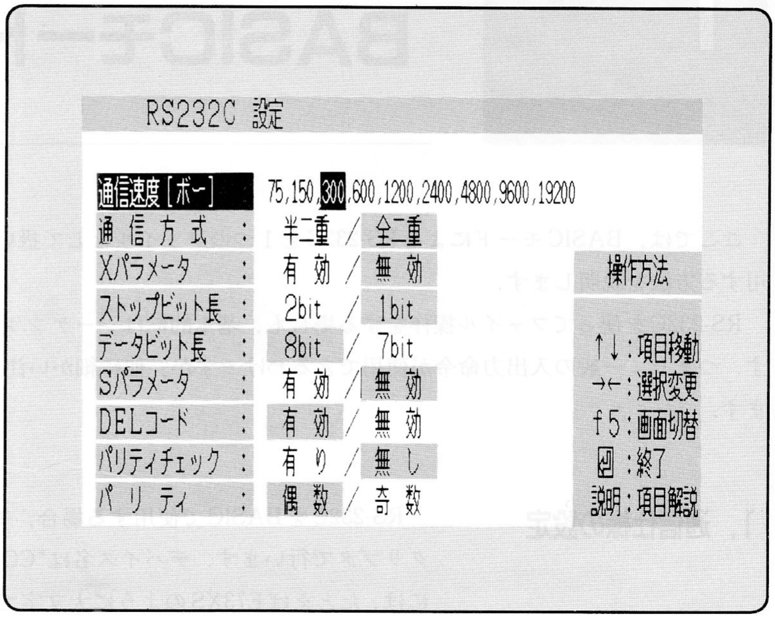
RS-232CをBASICで使用する場合，モードの指定はファイルディスクリプタで行います．デバイス名は"COM:"，ファイル名にあたる部分には，たとえばE73XSのように大文字でモードを指定します．モードについて，詳しくはリファレンスマニュアルのTERMコマンドの項を見てください．ただし，ボーレートはファイルディスクリプタで変更することはできません．また，RS-232Cをファイルとして扱う場合，いわゆるファイル名(ユーザが自由につけられる)というのは存在しません．

モード指定は省略が可能です．その場合は，セットアップモードによって設定された値が採用されます．初期設定は以下のようになっています．



注意　PC-8801MHの場合，拡張スロットタイプの項目はありません．

[f・5] を押すと，RS-232C設定の画面になります．



以下，第２画面（RS-232Cの設定）の各項目の意味を簡単に述べます．変更する場合は，第２章でご紹介した操作法により変更してください．

(1) 通信速度（ボー）

１秒間に送受信可能なビット数のこと．ボーレートが大きくなれば，短時間に大量のデータが送られる計算ですが，モデムや音響カプラを使用する際は，そのハード的制約にあわせて設定する必要があります．

(2) 通信方式（半二重／全二重）

全二重モードでは，データの送信，受信が同時に行えます．半二重モードでは，送信，受信のどちらかが行われているときには，１行分のデータの送信（受信）が終わるまで（CRコードが来るまで）次の受信（送信）は行えません．詳しくは第３節の資料「通信モード」を参照してください．

(3) Xパラメータ

Xパラメータを有効にすると，データ転送用バッファの４分の３までデータが入ると，自動的にホストコンピュータ（命令を送る側）にCTRL-Sコードを出力してデータ転送の一時停止を要求し，４分の１になるとCTRL-Qを出力して転送の再開を要求します．この機能は，ホストコンピュータがXパラメータを処理する機能を持つ場合に限って有効になります．

(4)　ストップビット長

RS-232Cを介してのデータの送受信は，通例，スタートビット（1ビット）＋データビット（7ビットまたは8ビット）＋パリティビット（1ビット）＋ストップビットの形式をとります．この最後のストップビットを1ビットにするか2ビットにするかがこのパラメータです．

(5)　データビット長

7ビットではアスキーキャラクタ，8ビットではANKキャラクタ（アスキーキャラクタにカタカナなどを加えたもの）が送れます．

(6)　Sパラメータ

Sパラメータを有効にすると，7ビットモードでカナを送受信することができます．つまり，SOコードがくると，それ以後のデータをカナとみなし，SIコードがくると，それ以後のデータを英数字とみなします．

(7)　DELコード

DELコードを受信した場合，その処理を行うか，無視するかを定めます．

(8)　パリティチェック

パリティとは，データが正確に送られたかどうかを検査するためのもの．具体的には，偶数パリティにするか奇数パリティにするかをあらかじめ決めておき，文字データを送るごとに，1ないし0を送り，文字データ＋パリティビットの和が決められたパリティになるよう操作します．

(9)　パリティ（偶数／奇数）

上記パリティチェックを，偶数で行うか奇数で行うかを決めます．

注意　本機とコンピュータとを接続する場合は，接続するコンピュータの仕様を十分に調べたうえで使用してください．

2. データファイルおよびアスキー形式のBASICプログラムの転送

基本的なサンプルプログラムを載せます．このようにRS-232Cも他のファイルと同じに扱えますが，通信速度を高速(1200ボー以上)にした場合は，データの取り出し速度より受信速度の方が速く，エラー(たとえば，"Buffer over flow"など)の出ることがありますので注意が必要です．

```
100 ' データファイルを送信する PROGRAM
110 '
120 INPUT "送信するファイルの名前は";FILE$
130 OPEN FILE$ FOR INPUT AS #1
140 OPEN "com:E73XS" FOR OUTPUT AS #2
150 '
160 WHILE NOT EOF(1)
170    LINE INPUT #1,A$
180    PRINT #2,A$
190 WEND
200 END
```

```
100 ' データファイルを受信する PROGRAM
110 '
120 INPUT "受信するファイルの名前は";FILE$
130 OPEN "com:E73XS" FOR INPUT AS #2
140 OPEN FILE$        FOR OUTUT AS #1
150 '
160 WHILE NOT EOF(1)
170    LINE INPUT #1,A$
180    PRINT #2,A$
190 WEND
200 END
```

3. ON COM GOSUB文(割り込み処理)

次に，RS-232Cによる割り込みを説明します．ON COM GOSUB文で割り込みが起こった場合の処理ルーチンを定義します．COM ONで割り込みを許可します．これ以後，RS-232Cから入力割り込みがあると，定義されている処理ルーチンへ制御を移します．RETURN文で中断したところから再開します．

下記のプログラムは送信，受信の両方に使用でき，割り込みによって受信されるようになっています．割り込みがない場合は160～180行の処理を繰り返し，キーボードからは1文字ずつ入力されるので，その文字をデータとして1文字ずつ送信し，また，入力がない場合も160～180行の処理を繰り返しますが，何も送信はしません．また，この間に割り込みが発生するとRECEVCHARのルーチンに飛んで，送信されてくる文字を画面に表示し，RETURN文で戻り，160～180行の処理を繰り返します．

```
100 FALSE=0
110 TRUE=NOT FALSE
120 ECHO=TRUE
130 OPEN "COM:E73XS" AS #1
140 ON COM GOSUB *RECEVCHAR
150 COM ON
160 C$=INKEY$
```

```
170 IF C$<>"" THEN PRINT #1,C$;:IF ECHO THEN PRINT C$;
180 GOTO 160
190 *RECEVCHAR
200 IF LOC(1)=0 THEN RETURN
210 N$=INPUT$(LOC(1),#1)
220 PRINT N$;
230 RETURN
```

4. ダイレクトモードでのプログラムの転送

ディスクなどの場合と同様，LOAD, SAVE(機械語ファイルについてはBLOAD，BSAVE)コマンドを用います．ただし，RS-232Cではファイルの終端を示すマークがなく，受信側(LOAD側)ではいつまでもコマンドから抜け出ません(ただし，プログラムは正常に受信されています)．したがって，時機をみて受信側で [停止] (あるいは [コントロール] + [C]) を押す必要があります．よって，離れた相手同士での通信では注意が必要です．

注意 入出力命令を使ってプログラムモードでプログラムファイルを送信する場合，他のファイル操作にも関連しますが，送受信されるプログラムファイルはアスキー形式でなければなりません．バイナリ形式のファイルや機械語ファイルでは，純粋な文字以外にコントロール文字"chr$(&HO)～chr$(&H21)"(「キャラクタコード表」参照)が含まれていることがあるため，画面の表示が乱れたり，プログラムが停止するなどの誤動作が起こる場合があるからです．

5. その他

(1) 使用可能な命令(一般のファイルとは意味の違う命令もあります)

● BASICで使用できる入出力命令

BLOAD	機械語プログラムを入力(受信)します．
BSAVE	機械語プログラムを出力(送信)します．
CLOSE	ファイルを閉じます．RS-232Cによる送受信を終了します．
EOF(関数)	バッファが空であると値を真とします．
GET	ファイル中のデータをバッファに読み込みます．
INPUT#/ LINE INPUT#	RS-232Cインタフェースファイルバッファから，データを入力(受信)します． INPUT#文のデータの区切りは，文字型では(,)(CR)(")，非文字型では(,)(CR)のいずれかです． LINE INPUT#文のデータの区切りは(CR)だけです．(CR)以外はすべてデータ

	として入力します.
INPUT$(関数)	ファイル番号で指定されたファイルから，指定した文字数の文字列を読み出します.
LOAD	プログラムを受信します(プログラムの受信が終了してもLOADコマンドは自動的に終了しません. 停止により強制的に終了させてください).
LOC(関数)	入力(受信)バッファ中に受信されているデータの文字数を知らせます.
LOF(関数)	バッファの残りバイト数を知らせます.
OPEN	通信の形式を定義し，RS-232Cインタフェースファイルバッファを開設します.
PRINT♯	データを出力(送信)します. 式と式の区切りには，(，)か(；)を使います.
PRINT♯ USING	フォーマット付きで，データを出力(送信)します.
PUT	バッファ中のデータをランダムファイルに書き出します.
WRITE♯	PRINT♯文と同様ですが，区切り記号としてコンマを必ず出力し，不要な空白の出力は行いません.
SAVE	プログラムを送信します.
WIDTH	RS-232Cボードに対し，指定されたサイズのデータを送出した後，CRコードを生成，送出します.

2 ターミナルモードでの通信

ここで説明するターミナルモードは，本機に内蔵されている簡単ターミナルプログラムを使う場合です．このターミナルプログラムは日本語を扱うことができません．

モデムを接続してターミナルモードを使用する場合には，以下の制限があります．

(1) 調歩同期モード(非同期モード)モデムだけが使用可能です．同期モードのモデムは使用できません．

(2) 全二重モデム(送信，受信それぞれの回線を使う形式)だけが使用可能です．
制御線を使って通信する半二重モデムは使用できません．

本機をターミナルとして使うことによって，遠く離れたホストコンピュータ(命令を送る側)から送られてきたデータを受け取ったり，電話回線を通し各種のコンピュータサービスを受けるなど，幅広く応用することができます．

注意 本機と他のコンピュータとを接続する場合は，接続するコンピュータの仕様を十分に調べたうえで使用してください．

1. ターミナルモードへの切り替え

本機を他のコンピュータ(ホストコンピュータ)のターミナル(端末)として使用するには，ターミナルモードに切り替えて使用するのが便利です．

ターミナルモードにするには，次の3つの方法があります．

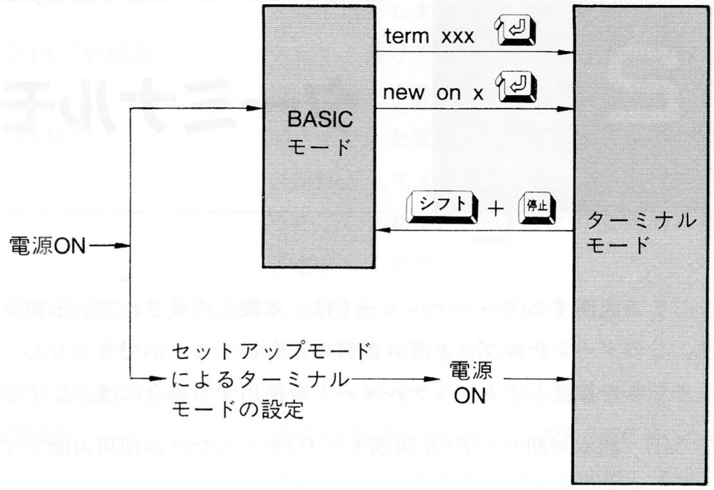
(1) BASICモードからTERM文によって切り替える

(2) BASICモードからNEW ON文によって切り替える

(3) あらかじめセットアップモードによって，ターミナルモードに設定しておく

本機をターミナルモードでしか使用しない場合は，(3)の方法が便利です．

また，本機がターミナルモードになっているときに，[シフト]を押しながら[停止]を押すと，BASICモードに切り替わります．



2. ボーレートの設定

ターミナルで使用するには，あらかじめセットアップモードによってボーレート(通信速度)を切り替えておきます．出荷時は300ボーにセットされていますから，ホストコンピュータの仕様に合わせて設定してください．設定画面については，前節をご参照ください．

3. TERM文を使用してBASICからターミナルモードに切り替える方法

BASICをスタートさせ，次のTERM文を入力します．

term "com:〈パリティ〉〈ビット長〉〈ストップビット〉〈Xパラメータ〉〈Sパラメータ〉"〔,〔〈通信モード〉〕〔,〈スタック長〉〕〕

各パラメータは，次のリスト中の希望のものを選び，右側に示されている英数を大文字で使用します(各パラメータの意味については，前節をご参照ください)．

〈パリティ〉：パリティチェックの指示を行う．

偶数パリティ(even parity) ………………… E

奇数パリティ(odd parity)…………………… O

パリティチェックなし(non parity)………… N

〈ビット長〉：データのビット長を示す．

7ビット……………… 7

8ビット……………… 8

〈ストップビット〉：ストップビットの長さを表す．

1ビット……………… 1

1.5ビット……………… 2

2ビット……………… 3

〈Xパラメータ〉：Xパラメータの有効，無効の切り替えを行う．

Xパラメータ無効……N

Xパラメータ有効……X

Xパラメータ省略……セットアップモードの値が採用される．

〈Sパラメータ〉：Sパラメータの有効，無効の切り替えを行う．

Sパラメータ有効……S

Sパラメータ無効……N

Sパラメータ省略……セットアップモードの値が採用される．

〈通信モード〉：全二重，半二重の指定を行う．

全二重……………………F

半二重……………………H

省略………………………セットアップモードの値が採用される．

〈スタック長〉：ターミナルモードでも，ホストコンピュータから送られてきたBASICのコマンドを実行することができますが，そのとき，使用する演算スタックの大きさを指定します．省略すると1024バイトが設定されます．

```
term "com:E72X",H ↵
```

偶数パリティ，データ長７ビット，ストップビット1.5ビット，Xパラメータ有効，半二重モードのターミナルモードになります．

TERM文のパラメータは，次のように省略することができます．

```
term "com:" ↵
term "  " ↵
```

この場合は，各パラメータともセットアップモードの値，または標準値(スタック長)が設定されます．

ターミナルモードになっている本機からホストコンピュータにBreak信号を送る場合は，[停止]を入力します．

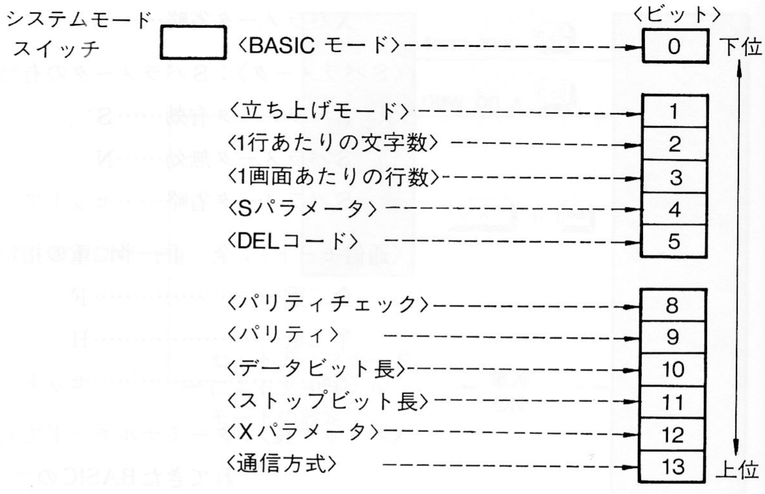
4. NEW ON文を使用してBASICモードからターミナルモードに切り替える方法

BASICをスタートさせ，次のNEW ON文を入力します．

```
new on〈式〉 ↵
```

NEW ON文はシステムモードスイッチ，セットアップモードによる各モードの設定をソフトウェアによって行うためのもので，〈式〉の値はシステムモードスイッチ，メモリスイッチの１個が１ビットに対応した(ONのとき１，OFFのとき０)16ビット表現の値を指定します．

NEW ON文によって指定できるのは，次のシステムモードスイッチおよびメモリスイッチです．その詳細については，次節および「リファレンスマニュアルNEW ON文」を参照してください．



（注）0，6，7，14，15ビットは意味を持たないので0にしてください．

例

new on 4866 ⏎

とすれば（BASICでは一般に数値は10進で表記されます），例では

term "com : E71X" , H ⏎

と同じターミナルモードになります（ただし，この場合，数値はメモリスイッチによって設定されている画面の桁数，行数によって異なってきます）．

5. セットアップモードでターミナルモードに切り替える方法（詳細は第2章参照）

(1) PC を押しながら，リセットボタンまたは電源スイッチを押すことにより，セットアップモードにします．

(2) 第1画面の第1項目，立ち上げモードをターミナル側にします．

(3) 第2画面の各パラメータについても，選択，設定します．

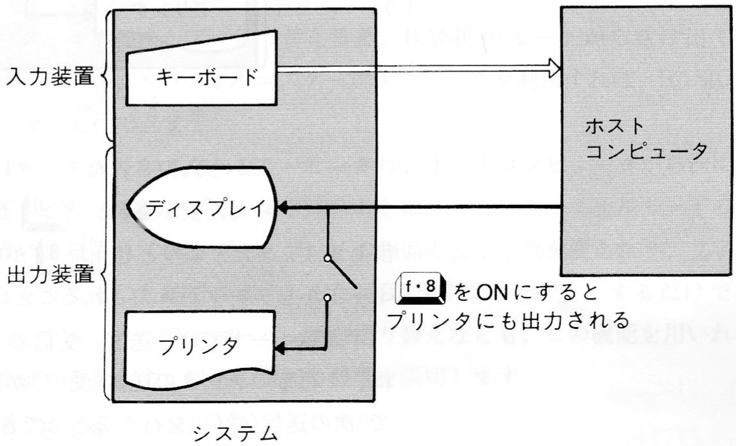
(4) セットアップモードを終了すると，最初からターミナルモードがスタートします．

3 資料　ターミナルモードの機能

1. 通信モード

本機をターミナルモードにしてデータの送受信を行うとき，通信モードは全二重モードまたは半二重モードで行います．

(1) 全二重モード

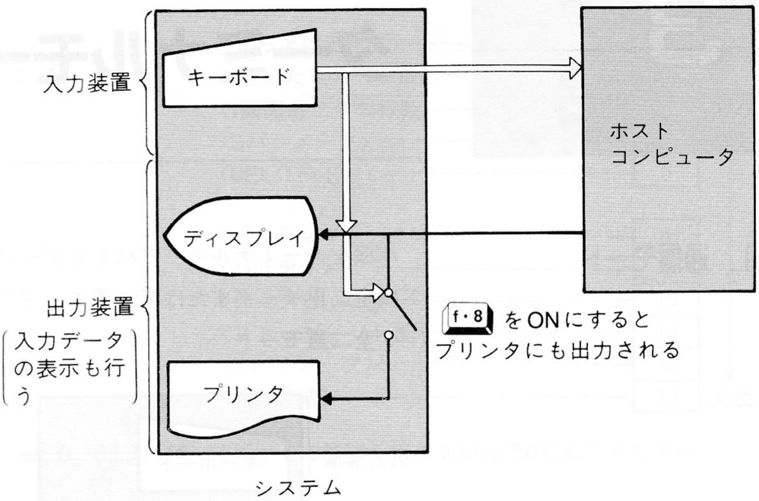


全二重モードでは，データの送受信を同時に行うことが可能です．

キーボードから入力されたデータはただちに回線に送出されますが，本機に接続されているディスプレイ，プリンタへは表示・出力されません．

受信データは自分のディスプレイに表示されますが，そのとき，f・8 がON(LPT enable)であれば，プリンタにも出力され，OFF(LPT disable)であれば出力されません．

(2) 半二重モード



キーボードから入力されたデータは，本機に接続されているディスプレイにいったん表示され，⏎ が押されると，データを相手に送出します．また，このとき f・8 がON(LPT enable)であれば，キーボードからデータが入力されるごとに，プリンタに出力します．

半二重モードでは，送信，受信のどちらかが行われているときには，1行分のデータの送信(受信)が終わるまで(CRコードがくるまで)次の送信(受信)を行うことはできません．

注意 本機のターミナルモード(BASICモードを含む)は，調歩同期全二重のモデムやカプラを対象に作られています．
したがって，半二重方式のモデムやカプラは使用することができません．

2. ファンクションキー

ターミナルモードに切り替えると，ファンクションキー f・1 ～ f・5 は，ターミナルモードになる前にBASICモードで定義されたものになりますが，ファンクションキーの f・6 ～ f・10 は次のように定義されます．

f・6 literal

このキーを入力すると，コントロールコードが実行されず，文字として表示されます．ただし，リターンとラインフィードは実行されます．

f・7 half/full

全二重，半二重の切り替えを行います．初期状態は，パラメータまたはメモリスイッチで設定されます．

f・8 LPT enable

画面への出力をプリンタへも出力します．もう1度入力すると，出力を停止します．

初期状態は出力しないモードに設定されています．

f・9 copy buffer

画面以外のバッファの内容をプリンタへ出力します．

f・10 LPT feed

プリンタに，1行フィードするコードを出力します．

3. スペシャルエスケープシーケンス

ターミナルモードに切り替えると，BASICのモードから抜け出てしまいますが，スペシャルエスケープシーケンスを利用すれば，BASICの命令を実行できます．

スペシャルエスケープシーケンスは，ホストコンピュータから次に示すようなエスケープコードを送ることによって，ターミナルモードの本機に接続されているディスプレイの画面の表示桁数を変えたり，文字の色を変えることができます．また，BASICの命令を実行するだけでなく，プリンタへのエコーバックの切り替えなども，この機能を用いれば簡単に行えます．以下に個別の機能を説明します．

(1) ESC@
ディスプレイ画面とプリンタへの出力(エコーバック)を開始する．

(2) ESC A
ディスプレイ画面とプリンタへの出力(エコーバック)を解除する．

(3) ESC "
本機のキー入力を有効にする．

(4) ESC ＃
本機のキー入力を無効にする．

(5) ESC !
ホストコンピュータへIDコードを出力する．

(6) ESC ＊
現在スクロールされている範囲をクリアする．

(7) ESC Y
現在のカーソル位置からスクロールされている最下位行までをクリアする．

(8) ESC □（□はスペースを意味します）
ターミナルのイニシャライズを行う(TERMコマンドを実行した直後と同じ状態にする)．

(9) ESC＝〈X-pos〉〈Y-pos〉

カーソル位置を指定する(20Hのバイアスがかかる).

(10) ESC T

現在カーソルがある行を1行消去する.

(11) ESC＞BASIC STATEMENT

BASICステートメントを実行する.

BASICの実行結果は，ターミナルとなっている本機に表示される.

(12) ESC . BASIC STATEMENT

BASICステートメントを実行する.

ただし，実行結果は表示されない.

(13) ESC＜BASIC STATEMENT

実行結果の表示はホストコンピュータ側だけ.

例 カーソルの位置を(3, 8)に移動させるには,

3→23H＝'♯'

8→28H＝'c'

ですから,

ESC＝♯C

とします.

4. リモートBASICプロトコル

前節スペシャルエスケープシーケンスで説明した機能の中で,

(11) ESC＞BASIC STATEMENT

(12) ESC . BASIC STATEMENT

(13) ESC＜BASIC STATEMENT

の3つを，リモートBASICプロトコル(remote BASIC protocol)といいます.

これによって，ホストコンピュータから，ターミナルとなっている本機のBASICの機能が利用できるようになります．ただし，この機能にはいくつかの制約があります.

(1) 使用できないBASICステートメント

リモートBASICプロトコルでは，メモリ内にプログラムを作成したり，メモリ内のプログラム変更，削除は行えません．また，メモリ内の変数エリアを直接扱うステートメント(たとえば，fre(0), clear, eraseなど)も実行できません．さらに，同じバッファを使って入出力を行うファイルをOPENさせることはできません(com:, cas:).

(2) 実行結果の表示

PRINTやPRINT USING文など実行結果を画面に表示させる命令では，表示をターミナル側に行うか，ホストコンピュータ側で行うかを選択できます．すなわち，エスケープコード(ESC)の後の符号(＞,

<)によって指定できるわけです．しかし，グラフィック命令はこの符号の指定にかかわらず，必ずターミナルとなっている本機に接続されているディスプレイに実行結果を表示します．

(3) 使用変数の制限

リモートBASICプロトコルでは，使用できる変数エリアが定まっています．また，一度使用すると，そのエリアは二度と別の変数には使えませんから，多くの変数に値を代入していくと，途中で代入できなくなってしまいます．リモートBASICプロトコルで使用する変数は，必要最小限にとどめてください．

例

● ターミナルにグラフを表示させる．

次のようにホストコンピュータからコードを送ると，ターミナルに鮮やかなグラフが簡単に表示できます．

```
ESC>screen 0,0
ESC>circle(100,50),100,7
ESC>a=100:b=100:c=200:d=50
ESC>line(100,50)-(a,b),7
ESC>line(100,50)-(c,d),7
ESC>paint(120,70),1,7
```

● BASICからターミナルモードに入った場合

次のようにすると，ターミナルになっている本機に接続されているフロッピーディスクにデータをセーブできます．

```
ESC>open "2:TEST1" for output as #1
ESC>print #1,100
ESC>close #1
```

第18章

デバッグ

プログラムが思ったとおりに動作しないことを「プログラムにバグ(虫)がいる」といい，それをなおすことをデバッグ(虫とり)といいます．
ここでは，BASICでプログラムを作ったときのデバッグの方法について説明します．

1 エラーメッセージを理解する

BASICは一般に，初心者にもわかりやすい言語だといわれていますが，それでも慣れないうちは正しくプログラムが組めるとは限りませんし，また，キー入力時に間違って打ち込むこともあります．

このような間違ったプログラムを実行すると，"Syntax error in 10"というようにエラーメッセージと，その発生行が表示されてプログラムが中断されます．

エラーメッセージの意味は「リファレンスマニュアル」巻末資料にまとめてありますので，これとプログラムのリストを見比べることで，エラーの原因の多くは知ることができます．

例

```
10 INPUT "A=",A
20 INPUT "B=",B
30 C=5*(A+(B-2)
40 PRINT "C=";C


run
A=10
B=20
Syntax error in 30
```

Syntax Errorとは文法エラーという意味ですが，そこで**list 30** ↵ として30行のリストを見ると，")"が1つ足りないことがわかりますから，そこを修正すればよいわけです．

このようにエラーメッセージの内容を理解すれば，多くの内容は修正できるものです．

注意 edit.□ ↵ としてもエラー行を表示させることができます．

上は，エラー発生行に原因があった例ですが，実際には発生行以外に原因のあることも多くありますから，その原因をつきとめることが必要です．

次のプログラムを実行してみましょう．

例

```
10 SCREEN 0:CLS 2
20 FOR X=0 TO 639
30    C=RND*7+1
40    LINE(X,100)-(X+10,199),C
50 NEXT
```

RUNさせますと，青から白までの7色の線が次から次へと描かれ，途中で"Illegal function call in 40"と表示されて中断してしまいます(もし中断しないときは，もう1度RUNさせてみてください)．

LIST 40 ⏎ として40行を表示させてみても，間違いがあるとは思われません．

このようなときには，この行の中で使用されている変数の内容を調べてみるとよいでしょう．

ダイレクトにPRINT X,C ⏎ と打ち込んでみましょう．

```
PRINT X,C
 XXX          8
Ok
```

×××はXの内容，0から639までの数字が表示されるはずです．

8というパレットコードは存在しませんから，このCの内容が不適当だということがわかります．

Cは30行でRNDという関数により乱数として作られています．RNDという関数は0から1未満の乱数を発生させますから，Cには1から7.9999……までの値が入ります．

LINE文のパレットコードには，その値が実数の場合，四捨五入されて与えられますから，エラーとなったのです．

このような場合は

```
30 C=INT(RND*7+1)
```

とすればよいわけです．

2　PRINT文で変数を調べる

エラーは起きないのに動作や結果が予想外であるという場合は，タイプミスなどで変数名が違っていたり，計算式が不適当だったりしたときによく起こります．

次の例を見てみましょう．

例

```
10 INPUT "ｽｳﾁ(1-300)=",S
20 JOG=S¥100+1
30 ON JOB GOTO 100,200,300
40 GOTO 400
50 '
60 PRINT "   1-100ﾉｼｮﾘｦｼﾏｽ.":END
70 PRINT "101-200ﾉｼｮﾘｦｼﾏｽ.":END
80 PRINT "201-300ﾉｼｮﾘｦｼﾏｽ.":END
90 PRINT "ｽｳﾁｶﾞｷﾃｲｶﾞｲﾃﾞｽ.":END
100 PRINT "   1-100ﾉｼｮﾘｦｼﾏｽ.":END
200 PRINT "101-200ﾉｼｮﾘｦｼﾏｽ.":END
300 PRINT "201-300ﾉｼｮﾘｦｼﾏｽ.":END
400 PRINT "ｽｳﾁｶﾞｷﾃｲｶﾞｲﾃﾞｽ.":END
```

このプログラムでは，どのような数値を入力してもすべて"キテイガイデス."と表示されます．

リストを調べてみても，特におかしいところも見当たらないようです．

そこで，ON-GOTO文で使われているJOBという変数の内容を調べてみましょう．

```
25 PRINT JOB

RUN
ｽｳﾁ(1-300)=150
 0
ｽｳﾁｶﾞｷﾃｲｶﾞｲﾃﾞｽ.
Ok
```

JOBが0では30行のGOTO以下の行番号には無視されますから，40行のGOTO 400に実行が移ったわけです．

このことから，JOBを計算している20行に原因があるらしいことがわかります．そこで20行を見ると，JOBのつもりがJOGになっていました．

では，修正して改めて実行してみましょう．

```
RUN
ｽｳﾁ(1-300)=150
 2
101-200ﾉｼｮﾘｦｼﾏｽ.
Ok
```

今度はうまくいったようですね.

では，100や200，300，−50などではどうでしょうか.

```
RUN
ｽｳﾁ(1-300)=200
 3
201-300ﾉｼｮﾘｦｼﾏｽ.
Ok
```

やはりおかしいですね.

結局，正しくは

```
20 JOB=INT((S-1)/100)+1
```

とすべきだったのです.

デバッグに際しては，予定外のデータを入力してみて，それらが正常に動作するかを調べてみることも必要なことです.

また，大きなプログラムではINPUT文の後にSTOP文やINPUT$文を入れて一時停止させると，ゆっくりと変数内容を確認できて便利です.

それには次のようにします.

PRINT A, B, C$: STOP ⏎

または

PRINT A, B, C$: S$=INPUT$(1) ⏎

STOP文はCONTコマンドで実行を再開できますし，INPUT$文なら何かキーを押すだけで続けられます.

デバッグが終了すれば，この行は消してしまってもよいのですが，もう1度使用するときのことを考えて，REM文でコメント文として残しておくとよいでしょう.

3 TRONコマンドを使ってプログラムの流れを見る

プログラムの実行順序がおかしい場合，TRONというコマンドを使ってプログラムの流れを見ることもできます．TRONコマンドは，行番号を表示しながら実行していくコマンドです．

たとえば次のプログラムを実行してみましょう．

```
10 A=0
20 A=A+.1
30 IF A=1 THEN PRINT A:END
40 GOTO 20

RUN
```

いつまでたっても終わりませんね．

では，TRONコマンドを使ってみましょう．

```
TRON
RUN
[10] [20] [30] [40] [20] [30] ······ [40] [20] [30] ····
```

20行から40行までがいつまでも繰り返されています．

このようにいつまでたってもループから抜け出せないことを「無限ループに入る」といいます．

これは30行に原因があります．

コンピュータの内部では数値を２進数で処理しているので，小数点以下の数値には誤差を生じてしまいます．実は，0.1は正確には0.1ではなく，10回足しても正確な１にはならずに30行は無視されていたのです．

こんな場合には，

```
30 IF A>=1 THEN PRINT A:END
```

とすれば，正しく動作します．

4 デバッグのテクニック

最後に，デバッグを行う際の注意すべき点，参考となる点をいくつかあげてみましょう．

(1) わかりやすいプログラムを書くように心がけましょう．

① 1行中に：で区切っていくつものステートメントをつなげることを「マルチステートメント」といいますが，あとでプログラムが見にくくなるので避けましょう．

```
10 INPUT "A=",A:INPUT "B=",B
20 C=A*B:PRINT C
```

↓

```
10 INPUT "A=",A
20 INPUT "B=",B
30 C=A*B
40 PRINT C
```

② ブロック（1つの動作を行うひとかたまりのプログラム）の先頭に，その内容を表すコメントをREM（または'）文で付けます．

また，ブロックとブロックの間にはREM文で空白行を入れます．

③ FOR～NEXTなどのループの内部は，2～3文字，字下げを行います．

```
10 FOR I=32 TO 255 STEP 16
20    FOR J=0 TO 15
30       C=I+J
40       PRINT CHR$(C);
50    NEXT J
60 NEXT I
```

(2)　演算子の優先順位に注意しましょう．

式が複雑なときや，自信のないときには，とりあえず(　)を付けましょう．

(3)　整数化する際のまるめ方に注意しましょう．

N_{88}-BASICでは，小数点以下は四捨五入されます．

(4)　コロン(：)とセミコロン(；)，ピリオド(．)とコンマ(，)，ゼロ(Ø)とオー(O)，エル(l)と数字のいち(1)，ビー(B)と(8)が間違っていないかを注意しましょう．

(5)　必要に応じてPRINT文やSTOP文，INPUT$文，TRONコマンドなどを用い，プログラムの実行順序や変数の内容を確認しましょう．

(6)　ファイルの出力を行うプログラムの場合，デバッグ中は画面に出力しましょう．

```
10 DV$="SCRN:"
20 OPEN DV$ FOR OUTPUT AS #1
```

デバッグが終了したら，10行を

```
10 DV$="TEST.DAT"
```

と書き換えます．

また，ファイルからの入力のかわりにキーボードから入力を行うこともできます．

```
10 DVIN$="KYBD:"
20 OPEN DVIN$ FOR INPUT as #1
```

(7)　ON ERROR GOTOを使ったプログラムをデバッグする場合，ON ERROR GOTOが実行される状態でエラーが形を変えて現れることがあるので，デバッグがやりにくくなります．

デバッグ中はON ERROR GOTO文をコメント文(REMを使用)に変更しておきましょう．

```
100 'ON ERROR GOTO 160
110 INPUT "X :";X:INPUT "Y :";Y
130 IF X=0 AND Y<0 THEN ERROR 250
140 Z=X^Y:PRINT " Y":PRINT "   X=";Z
150 GOTO 120
160 IF ERR=250 THEN PRINT "ﾃｲｷﾞ ｻﾚﾏｾﾝ."
170 IF ERR=6   THEN PRINT "ｵｰﾊﾞｰ ﾌﾛｰ ﾃﾞｽ."
180 RESUME 120
Ok
RUN
X :? 2
Y :? 10
 Y
  X= 1024
Undefined line number in 150
Ok
```

第19章

機械語モニタ

この章は，本機で直接機械語を扱う場合に使用する機械語モニタについて説明します．

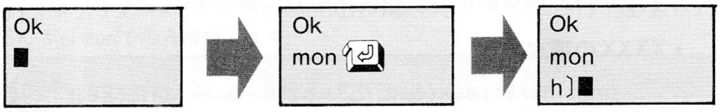
本機には，独立したI/Oコントロールルーチンを持つ強力な機械語モニタが準備されています．この機械語モニタの主な特長は次のとおりです．

1．ミニアセンブラ機能を持っている．
2．数値を16進数，8進数のどちらでも入力，表示が可能．
3．I/Oポートへの入出力が可能．
4．逆アセンブル機能を持っている．
5．CPUのレジスタの値の表示，変更が可能．
6．カセットテープにもフロッピーディスクにもロード，セーブが可能(PC-8801MHでカセットテープを使う場合は，CMTインタフェースボードが必要)．
7．スクリーンエディタによるメモリの内容の変更ができる．
8．コマンドの入力形式を知りたいときのために，HELPコマンドが用意されている．

1 機械語モニタの動かし方

機械語モニタは，BASIC モードから機械語モニタモードに切り替えて使用します．

1. BASICモードから機械語モニタモードへの切り替え方法

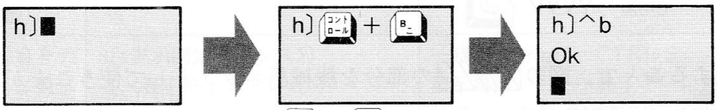


BASICコマンドが入力可能．

MON コマンドを入力．

機械語モニタモードになり，h]が表示される．
機械語モニタのコマンド入力が可能．

2. 機械語モニタモードからBASICモードへ戻るための方法



機械語モニタのコマンドが入力可能．

［コントロール］と［B］とを同時に押す．

BASICモードになり，Okが表示されて，次の行にカーソルが現れる。
BASICコマンドが入力可能．

2 機械語プログラムのメモリ配置

BASICを使っているときのメモリマップは下図のようになっています．

XXXX，ZZZZの番地は，選ばれているBASICのモードなどによって変わります．機械語プログラムをメモリ上に書き込む前に，次のようにして番地を調べてください．

- **XXXXの調べ方**

  print hex$(peek(&HE7E9)*256+peek(&HE7E8)) ⏎

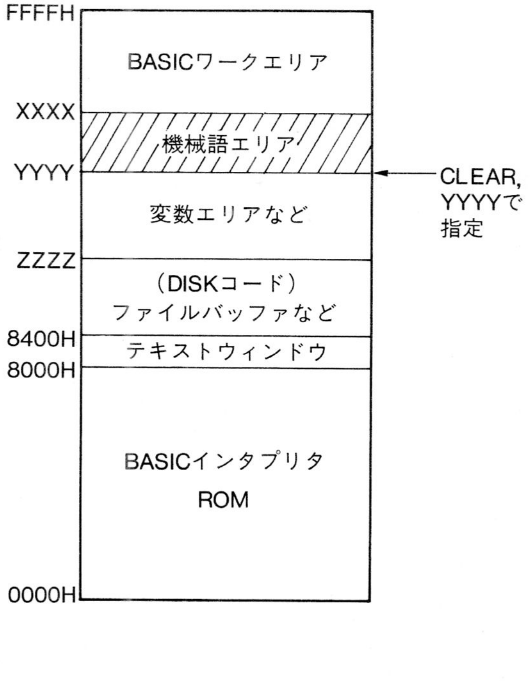
- **ZZZZの調べ方**

  print hex$(peek(&HEB1C)*256+peek(&HEB1B)) ⏎

機械語のプログラムを作る場合は，CLEAR命令で機械語エリアを確保しておかなければなりません．

clear ,YYYY ⏎

とすることで，図の▨の部分を機械語エリアとして使うことができるようになります．それ以外の部分はBASICが使っていますので，その内容を変更すると，BASICモードに戻ったときの動作が保障されません．



3 コマンドの種類

機械語モニタのコマンドには，表に示すように22種類のコマンドがあります(命令は大文字でも小文字でも入力できます).

*　CMTインタフェースボード(PC-8801MHの場合)とデータレコーダが必要です.
**　DISK Versionでのみ機能します.

コマンド	意味	機能	入力形式	オペレーション
A	アセンブル	入力した1行をアセンブルする.	a〔〈格納開始アドレス〉〕	インサート………INS デリート…………DEL 終了………………STOP, CTRL-C
B	ベース	数値の表現形式を変える. (8進↔16進)	bqまたはbh	―――
D	ダンプメモリ	メモリの内容をディスプレイに表示する.	d〔〈表示開始アドレス〉〕 〔,〈表示終了アドレス〉〕	一時中断…………CTRL-S 表示再開…………終了キー以外 終了………………STOP, CTRL-C
E	エディットメモリ	スクリーンエディタの機能を用いてメモリの内容を変更する.	e〔〈開始アドレス〉〕	カーソル移動……カーソル移動キー ロールアップ……ROLL UP ロールダウン……ROLL DOWN 終了………………STOP, ESC
F	フィルメモリ	メモリの内容を定数で埋めていく.	f〈開始アドレス〉, 〈終了アドレス〉,〈定数〉	―――
G	ゴー	ユーザプログラムを実行する.	g〔〈実行開始アドレス〉〕 〔,〈ブレイクポイントアドレス1〉〕 〔,〈ブレイクポイントアドレス2〉〕	―――
I	インプット	I/O ポートの値を読み込む.	i〈ポートアドレス〉	―――
L	ディスアセンブル	機械語を逆アセンブルする.	l〔〈逆アセンブル開始アドレス〉〕 〔,〈逆アセンブル終了アドレス〉〕	一時中断…………CTRL-S 再開………………終了キー以外 終了………………STOP, CTRL-C
M	ムーブメモリ	ある範囲のメモリの内容を他のアドレスのメモリ領域へ移す.	m〈転送元・先頭アドレス〉, 〈転送元・最終アドレス〉, 〈転送先・先頭アドレス〉	―――
O	アウトプット	I/O ポートへデータを出力する.	o〈ポート・アドレス〉,〈データ〉	―――

コマンド	意味	機能	入力形式	オペレーション
P	プリンタスイッチ	プリンタへの出力をコントロールする.	p	————
R*	リードテープ	カセットテープからデータをロードする.	r〔〈ファイル名〉〕	————
S	セットメモリ	メモリにデータをセットする.	s〔〈開始アドレス〉〕	セット……………スペース, リターン アドレス後退……CTRL-B 終了………………リターン
TM	テストメモリ	メモリをテストする.	tm	————
V*	ベリファイテープ	カセットテープの内容と，メモリの内容を比較する.	v〔〈ファイル名〉〕	————
W*	ライトテープ	メモリの内容をカセットテープにセーブする.	w〔〈レジスタ名〉〕 〈セーブ開始アドレス〉 〈セーブ終了アドレス〉	————
X	イグザミンレジスタ	CPU のレジスタの値を調べ，変更する.	x〔〈レジスタ名〉〕	次レジスタ………スペース 終了………………リターン
[説明]または CTRL-A	ヘルプ	コマンドとそのパラメータの形式をディスプレイに表示する.	CTRL-Aまたは[説明]	————
CTRL-B	リターン	BASIC モードへ復帰する.	CTRL-B	————
CTRL-D**	ダンプディスク	フロッピーディスクの内容をディスプレイに表示する.	CTRL-D 〈ドライブ♯〉〔,〈サーフェス♯〉〕, 〈トラック♯1〉,〈セクタ♯1〉 〔,〈トラック♯2〉,〈セクタ♯2〉	————
CTRL-R**	リードディスク	フロッピーディスクから，データをロードする.	CTRL-R 〈ドライブ♯〉〔,〈サーフェス♯〉〕, 〈トラック♯〉,〈セクタ♯〉, 〈開始アドレス〉,〈終了アドレス〉	————
CTRL-W**	ライトディスク	フロッピーディスクへデータをセーブする.	CTRL-W 〈ドライブ♯〉〔,〈サーフェス♯〉〕, 〈トラック♯〉,〈セクタ♯〉, 〈開始アドレス〉,〈終了アドレス〉	————

4 コマンド使用上の規則

1. 入力文字

機械語モニタのコマンドの英文字は，大文字でも小文字でも入力できます．

2. エディト機能

機械語モニタでは，BASICとは異なって，スクリーンエディタの機能がありません(ただし，Eコマンドを参照)．また，[後退]を使って入力した文字を消して修正することもできません．誤ったコマンドを入力したことに気づいたときは，[停止]または[コントロール]＋[C]で誤ったコマンドを取り消したのち，改めて正しいコマンドを入力してください．

3. エラー表示

機械語モニタでは入力できるキーは少数に限られていて，誤ったキーを入力すると"？"が表示されます．

4. 数値形式

機械語モニタで扱う数値形式には，16進数モードと8進数モードがあります．この切り替えはBコマンドで行います．システムが起動した直後は16進数モードになっています．

たとえば，16進数を入力するときに，BASICのように&Hを付けて入力する必要はありません．

(1) 16進数モード

アドレス　　：4桁　0～FFFF

メモリの内容：2桁　0～FF

(2) 8進数モード

アドレス　　：6桁　0～177777

メモリの内容：3桁　0～377

アドレスの場合は，入力された数値の終わりから16ビットが有効になります．メモリの内容の場合は，終わりから8ビットが有効になります．

また，機械語モニタの各コマンドでアドレスやメモリの内容を入力する場合，加減の演算機能が使用できます(ただし，カッコや累乗は使えません).

```
h〕s9000＋100 ↵
9100 00―
```

5. カセットテープのスピード

機械語モニタによってカセットテープを使用するには，Rコマンド，Vコマンド，Wコマンドを用います．このとき，ファイル名の指定の方法により転送速度が決まります．

R〔1: / 2:〕〔〈ファイル名〉〕

W〔1: / 2:〕〔〈ファイル名〉〕,〈セーブ開始アドレス〉,〈セーブ終了アドレス〉

V〔1: / 2:〕〔〈ファイル名〉〕

ファイル名	転送速度
〔〈ファイル名〉〕 1：〔〈ファイル名〉〕	1200bps
2：〔〈ファイル名〉〕	600bps

6. 機械語モニタモードでのディスプレイ画面

- BASICモードから機械語モニタモードへ切り替えた直後は，その直前のWIDTH文によって設定された状態が引きつがれます．
- テキスト画面のスクリーンウィンドウによっては，Eコマンドは動作しない場合があります．

  このときは，機械語モニタモードに入る前に，

  console 0,〈17以上の値〉↵

  を実行しておいてください．

A（アセンブル）

Intel 8080ニーモニックで入力した1行のテキストをアセンブルし，できた機械語をメモリに格納します．そのために，Aコマンドは，

```
a〔〈格納開始番地〉〕
```

のように，普通は機械語の格納を開始する番地を指定します（格納開始番地の指定を省略した場合は，その前に実行したAコマンドの実行終了番地の次の番地となります）．

```
h]a9000 [↵]
```

Aコマンドを入力すると，格納開始番地を表示して，ニーモニックの入力待ちになります．そこで，Intel 8080ニーモニック命令を入力し，[↵]を入力すると，その1行をアセンブルし，できた機械語をアドレスのすぐ横に表示し，そのアドレスに格納します．そして，その次の行に，次に機械語を格納する番地を表示し，ニーモニック命令の入力待ちになります．

```
h]a9000 [↵]
9000 3E 01 mvi a,1 [↵]
```

入力したニーモニック命令にエラーがあって"？"が表示された場合には，もう1度アドレスを指定しなくても，コマンドだけを入力すれば，エラーを起こしたときのアドレスから実行を開始します．

```
h]a9000 [↵]
9000 3E 01 mvi a,1 [↵]
9002       mvl b,1 [↵]
```

このアセンブラはすべて，[↵]が入力されるまで読み込みを続けます．また[削除]が使えますから，間違えて入力した文字を訂正できます．このアセンブラから抜け出すには，アドレスが表示されたあと[↵]を入力するか，[停止]または[コントロール]＋[C]を入力します．

このアセンブラで使用できるニーモニックは，Intel 8080ニーモニック命令とZ80ニーモニック命令の中のレラティブジャンプ命令です．Z80のレラティブジャンプ命令の場合は，絶対アドレスを入力すれば，アセンブラが相対アドレスに変換してくれます．

Z80のレラティブジャンプ命令の入力形式は，下のとおりです．

Z80	モニタで入力するとき
jr 〈アドレス〉	jmpr 〈アドレス〉
jr z, 〈アドレス〉	jrz 〈アドレス〉
jr nz, 〈アドレス〉	jrnz 〈アドレス〉
jr c, 〈アドレス〉	jrc 〈アドレス〉
jr nc, 〈アドレス〉	jrnc 〈アドレス〉
djnz 〈アドレス〉	djnz 〈アドレス〉

B (ベース)

数値を取り扱う形式を設定するコマンドです．数値をキーボードから入力したり，ディスプレイに表示する場合に，16進数として扱うか 8 進数として扱うかを決めます．BASICモードからモニタへ切り替えたときは16進数モードになっていて，コマンド入力待ちになると，

h〕

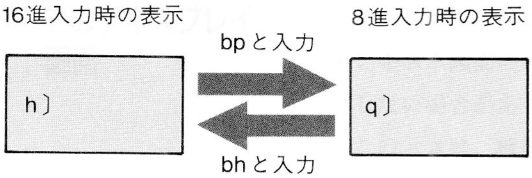
が表示されます．8 進モードに切り替えるには，bq ⏎ と入力します．

8 進数モードになると"q〕"が表示されます．

h〕bq ⏎
q〕

再び16進数モードへ切り替えるにはbh ⏎ を入力します．

q〕bh ⏎
h〕



D (ダンプメモリ)

メモリの内容をディスプレイに表示するコマンドです．16進モードになっている場合は，その数に対応するアスキー文字も表示されます．さらに，プリンタスイッチでプリントモードが指定されていれば，結果がプリンタへも出力されます（プリンタスイッチおよびプリントモードについては，「P（プリンタスイッチ）」を参照してください）．Dコマンドは次の形で入力されます．

d〔〈表示開始アドレス〉〕〔,〈表示終了アドレス〉〕

ですから，実際の入力は次の形になります．

(1) d9000, 9030 ⏎
(2) d9000 ⏎
(3) d, 9030 ⏎
(4) d ⏎

(1)の場合が標準で，9000H番地から9030H番地までのメモリの内容が表示されます．

(2)のように〈表示開始アドレス〉だけ指定した場合は，〈表示開始アドレス〉から16バイト分のメモリの内容が表示されます．

逆に(3)のように〈表示終了アドレス〉だけ指定した場合は，その前に実行されたDコマンドの最終のアドレスの次のアドレスから〈表示終了アドレス〉までのメモリの内容を表示します．

(4)のように〈表示開始アドレス〉も〈表示終了アドレス〉も指定しない場合は，このコマンドが入力される前に実行されたDコマンドの最終のアドレスの次のアドレスから16バイト分のメモリの内容が表示されます．

Dコマンドを実行中に，その実行を中止したい場合は，[停止]または[コントロール]＋[C]を入力します．画面にメモリの内容を表示している場合，途中で表示を一時停止させるには，[コントロール]＋[S]を入力します．再び表示を再開させる場合には，[停止]と[コントロール]＋[C]以外のキーのどれかを入力します．

また，Dコマンド終了後に [↵] を入力すると(4)と同じ動作を行います．

E（エディットメモリ）

Eコマンドは，スクリーンエディタの機能を用いて，スクリーンに表示されたメモリの内容を変更するコマンドです．Eコマンドは次の形式で入力します．

e〔〈開始アドレス〉〕

〈開始アドレス〉を省略した場合は，最後にEコマンドで内容を変更したメモリのアドレスが〈開始アドレス〉となります．

Eコマンドを実行して，メモリの内容をディスプレイに表示した後は，カーソル移動キー [↑] [↓] [←] [→] と [⇩] [⇧] を使って変更したいメモリの内容のところへカーソルを移動させ，新しい数値を入力します．画面上で変更された数値のとおりに，メモリの内容も変更されています．エディットメモリを終了するときは，[停止]，[ESC]，または [コントロール] + [C] を入力します．

F（フィルメモリ）

アドレスで指定した範囲のメモリの内容を，指定した定数の値に置き換える命令です．コマンドの形式は次のようになっています．

f〈開始アドレス〉,〈終了アドレス〉,〈定数〉

Fコマンドでは，各パラメータを省略することはできません．コマンドが正常に実行されたかどうかは，Dコマンドを用いて確認します．

```
h]f9000,9010,0 [↵]
h]d9000,9010 [↵]
9000 00 00 00 00 00 00 00 00
9008 00 00 00 00 00 00 00 00
9010 00
h]
```

G（ゴー）

指定したアドレスにジャンプして，そこからプログラムの実行を開始します．このとき，2つまでブレイクポイントアドレスを指定できます．ブレイクポイントアドレスとは，プログラムの実行中，実行アドレスがブレイクポイントアドレスとなったとき，実行を中止して，コマンドの入力待ち状態にさせるアドレスです．

g〔〈実行開始アドレス〉〕〔,〈ブレイクポイントアドレス#1〉〕〔,〈ブレイクポイントアドレス#2〉〕

〈実行開始アドレス〉を省略した場合，最後に実行したブレイクポイントアドレスが〈実行開始アドレス〉になります．ブレイクポイントアドレスで実行が中断された場合は，CPUのレジスタの値は保存されます．

＊ブレイクポイントアドレスは，指定したアドレスがRAMにあるときだけ有効になります．

I（インプット）

入力ポートの値を読み，画面に表示します．ポートアドレスを入力した後，スペースを入力すれば，その行に入力ポートの値を表示し，[↵] を入力した場合は，次の行に入力ポートの値が表示されます．

i〈ポートアドレス〉[↵]
（または，スペースキー）

（「O（アウトプット）」の項を参照してください．）

L（ディスアセンブル）

メモリイメージをIntel 8080ニーモニックに逆アセンブルして表示するコマンドです．さらに，Z80のレラティブジャンプ命令も使用できますが，アドレスは，絶対アドレスで表示されます．Lコマンドの形式は次のようになっています．

l〔〈逆アセンブル開始アドレス〉〕〔,〈逆アセンブル終了アドレス〉〕

〈逆アセンブル開始アドレス〉が省略された場合は，最後に逆アセンブルされたアドレスの次のアドレスが〈逆アセンブル開始アドレス〉になります．また，〈逆アセンブル終了アドレス〉が省略された場合は，〈逆アセンブル開始アドレス〉＋15が〈逆アセンブル終了アドレス〉となります．

Lコマンドの実行中に実行を中断したい場合は，[停止]または[コントロール]＋[Cそ]を入力します．実行を一時中断したい場合は，[コントロール]＋[Sと]を入力します．実行を再開するには[停止]，[コントロール]＋[Cそ]以外の任意のキーを入力します．また，Lコマンドで表示されるニーモニックのオペランドは，現在指定されている数値モードで表示されます．

さらにLコマンド終了後に[↵]を入力すると，〈逆アセンブル開始アドレス〉と〈逆アセンブル終了アドレス〉を省略したときと同じように，〈逆アセンブル終了アドレス〉から16バイト分のメモリイメージを逆アセンブルします．

M（ムーブメモリ）

Mコマンドは，メモリ内で，ある位置から別の位置へメモリ領域のブロック転送を行います．Mコマンドは次の形式で入力します．

m〈転送するメモリ領域の先頭番地〉,〈転送するメモリ領域の最終番地〉,〈転送先の先頭番地〉

たとえば，9000H番地から9100H番地のメモリ内容をA000H番地からA100H番地へ転送する場合は，

h〔m9000, 9100, a000 [↵]

と入力します．この後，Dコマンドを実行すれば，転送できたことを確認できます．

O（アウトプット）

このコマンドは，指定したポートアドレスにデータを出力します．Oコマンドの入力形式は，

o〈ポートアドレス〉,〈データ〉

となります．

IコマンドおよびOコマンドは，自作のボードを機械語プログラムでコントロールする場合など，特殊な用途に使用するコマンドです．

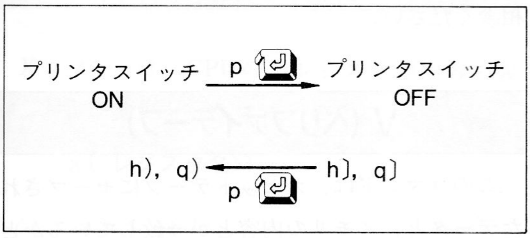
P（プリンタスイッチ）

Dコマンド，Lコマンド，CTRL-Dコマンドを実行した結果を，プリンタに出力するかどうかを決定します．

モニタに入ったときは，プリンタスイッチはOFFになっていて，Dコマンド，Lコマンド，CTRL-Dコマンドを実行しても，結果はディスプレイに表示されるだけです．このときPコマンドを入力すると，プリンタスイッチはONになり，以後，実行されるDコマンド，Lコマンド，CTRL-Dコマンドの結果はプリンタにも出力されます．

プリンタスイッチがONになると，プロンプト（入力促進記号）は**h**），あるいは**q**）になります．

プリンタスイッチをONから再びOFFにするには，再度Pコマンドを入力します．



＊モニタを使用しているときも 画面コピー は使用できます．画面のハードコピーを取る場合は 画面コピー を使用してください．

R（リードテープ）

カセットテープからデータをロードするコマンドです.* Rコマンドは，次の形式で入力します．

r〔〈ファイル名〉〕

ファイル名を指定した場合は，カセットテープのデータのファイル名をチェックし，指定したファイル名が見つかると，そのデータをメモリにロードします．このとき，指定したファイル名以外のファイル名が見つかった場合は，

Skip XXXX

とディスプレイに表示し，指定したファイル名が見つかるまで読み続けます．指定したファイル名が見つかった場合は，

Found XXXX

とディスプレイに表示し，その下にロードしているデータを表示します．

ファイル名を指定しない場合は，最初に見つかったファイルをロードします．

ファイル名は6文字以下です．6文字を超えた場合は，最初の6文字が有効になります．

＊PC-8801MHでこのコマンドを使用するには，データレコーダと別売のCMTインタフェースボードが必要です．

S (セットメモリ)

Sコマンドは，メモリの内容を確かめたり，変更するコマンドです．Sコマンドは次の形式で入力します．

s〔〈開始アドレス〉〕 ↵
（または，スペースキー）

Sコマンドで，〈開始アドレス〉を指定した後，↵ を入力すると，次の行にアドレスとその内容を表示し，数値の入力待ちになります．スペースキーを入力すると，改行せずにメモリの内容を表示し，数値入力待ちになります．

入力する数値は，16進モードならば2桁の16進数を入力しますが，2桁以上の数値を入力した場合は，最後の2桁が有効となります．数値を入力した後スペースキーを入力すれば，メモリに値をセットし，次のアドレスのメモリの内容を表示します．また，数値を入力した後，↵ を入力すれば，新しい値がセットされ，コマンド入力待ちになります．もし，メモリの内容を変更しないならば，数値を入力せずにスペースキーを入力すれば，次のアドレスに移ります．アドレスを1つ前に戻したい場合は，[コントロール]＋[B] または [←] を入力します．

TM (テストメモリ)

N₈₈-BASICで使用されるユーザーズメモリ64Kバイトと，グラフィック用VRAM48Kバイトをテストするコマンドです（N₈₈-日本語BASICで使われる拡張RAM128Kバイトのテストは行いません）．テストは次の順序で行います．

(1) モニタに入る前に
　　screen 0,2 ↵
　を入力する．
(2) モニタへ入る．
　　mon ↵
(3) tm ↵
(4) 次の順でチェックします．
　① テキストVRAM
　② 変数・ワークエリア
　③ グラフィック用VRAM
　④ テキストエリア

このテストには，V2モードでは約10分間，V1標準モードでは約30分間かかります．テストが無事に終了すると，

　test complete!

というメッセージが現れますから，リセットボタンを押してください．

エラーが起きた場合，①で起きたときはブザー（beep音）が鳴り続けます．②以後の場合は，エラーが起きた番地を表示します．

TMコマンドでエラーが発生した場合は，お買い求めの販売店，あるいは最寄りのBit-INNにご相談ください．

V (ベリファイテープ)

このコマンドは，カセットテープにセーブされたデータと，メモリの内容とが一致しているかどうかを比較するコマンドです*．Vコマンドは，次のように入力します．

v〔〈ファイル名〉〕 ↵

動作はRコマンドとほぼ同じで，〈ファイル名〉を指定した場合は，指定したファイルとメモリの内容を比較し，〈ファイル名〉を指定しなかった場合は，最初に見つかったファイルと，メモリの内容を比較します．比較した結果，カセットテープの内容とメモリの内容が一致した場合はコマンド入力待ち状態となり，一致しなかった場合は"？"を表示してコマンド入力待ちになります．

＊PC-8801MHでこの命令を使用するには，データレコーダと別売のCMTインタフェースボードが必要です．

W（ライトテープ）

メモリの内容をカセットテープへセーブするコマンドです.* Wコマンドは次の形式で入力します.

w〔〈ファイル名〉〕,〈セーブ開始アドレス〉,〈セーブ終了アドレス〉 [↵]

8000H番地から，8500H番地の内容を"abc"というファイル名でカセットテープにセーブする場合，

h]wabc, 8000, 8500 [↵]

*PC-8801MHでこの命令を実行するには，データレコーダと別売のCMTインタフェースボードが必要です.

X（イグザミンレジスタ）

Xコマンドは，CPUの全レジスタの内容の表示，および変更を行うコマンドです.

x〔〈レジスタ名〉〕

〈レジスタ名〉は，

a, b, d, h, a', b', d', h', s, p, x, y, i

です．a'のようにダッシュ記号が付いているレジスタ名は，裏レジスタを指します.

〈レジスタ名〉を指定しない場合は，全レジスタの内容を表示して実行を終了します．〈レジスタ名〉を指定した場合は，そのレジスタの内容を表示し，新しい値の入力待ちになります．新しい値を入力して [↵] を入力すると，コマンド入力待ち状態になり，スペースキーを入力すると，次のレジスタの処理へと移ります．新しい値を入力しない場合も，[↵] を入力すればコマンド入力待ち状態となり，スペースキーを入力すると，次のレジスタの処理へと進みます.

各レジスタは，Iレジスタを除いて，すべてのレジスタは16bitのペアレジスタとして扱われます．ですから，8bit分のレジスタだけ変更する場合は，変更しない残りの8bit分のデータも入力してください．また，フラグレジスタでフラグを立てないときは－を入れます.

CTRL-A（ヘルプ）

[説明] または [コントロール] ＋ [A] を入力すると，モニタで使用できるコマンドの種類とそのコマンドのパラメータがディスプレイに表示されます.

CTRL-B（リターン）

[コントロール] ＋ [B] を入力すると，BASICモードに戻ります.

CTRL-D（ディスクダンプ）

CTRL-Dは，フロッピーディスクの内容をディスプレイに表示させるコマンドです.

CTRL-D〈ドライブ#〉〔,〈サーフェス#〉〕,〈トラック#1〉,〈セクタ#1〉〔,〈トラック#2〉,〈セクタ#2〉〕

〈ドライブ#〉〔,〈サーフェス#〉〕,〈トラック#1〉,〈セクタ#1〉
↑
表示開始セクタの指定

〔,〈トラック#2〉,〈セクタ#2〉〕
↑
表示終了セクタの指定

〈トラック#2〉,〈セクタ#2〉を指定しない場合は，

〈表示開始セクタ〉＝〈表示終了セクタ〉

となり，表示開始セクタだけ表示します.

CTRL-Dコマンドを実行中，実行を中止したい

場合は[停止]または[コントロール]＋[C]を入力します．画面への表示を一時中断したい場合は[コントロール]＋[S]を入力し，再開するときは[停止]および[コントロール]＋[C]以外のキーを入力します．

〈サーフェス＃〉は，両面倍密度のフロッピーディスクユニットを使用しているときに指定します．ただし，モニタでは両面にまたがってアクセスすることはできません(CTRL-R，CTRL-Wも同じです)．

ディスク装置を内蔵していないタイプのPC-8801FHでは，この命令は使用できません．

- CTRL-D, CTRL-R, CTRL-Wコマンドで，5.25インチ両面高密度ミニフロッピーディスクのサーフェス0，トラック0を指定するときは，セクタ＃は奇数番号だけが使用できます．
- サーフェス0，トラック0は，他の部分とは異なり1セクタが128バイトから成りますから，一度に2セクタ分ずつ読み書きします．
- 8インチフロッピーディスクでは，CTRL-D, CTRL-R, CTRL-Wコマンドをトラック0に対して行うことはできません．

CTRL-R (リードディスク)

CTRL-Rコマンドは，フロッピーディスクからメモリへデータをロードするコマンドです．

CTRL-R〈ドライブ＃〉〔,〈サーフェス＃〉〕,〈トラック＃〉,〈セクタ＃〉,〈開始アドレス〉,〈終了アドレス〉

このコマンドは，〈開始アドレス〉と〈終了アドレス〉で指定されたメモリの領域に格納するデータを，〈ドライブ＃〉〔,〈サーフェス＃〉〕，〈トラック＃〉，〈セクタ＃〉で指定されたセクタからロードします．ですから，大きなメモリ領域にデータをロードする場合は，複数のセクタ，トラックからデータをロードします．

```
h] ^r1,0,9,3, C000,C00f
h]
```

ドライブ1の第9トラックの第3セクタから16バイトのデータをロードしてきて，C000H番地からC00FH番地に格納します．このとき，メモリ領域のC010H～C0FFH番地の内容は変更されません．

- 8インチフロッピーディスクでは，CTRL-D, CTRL-R, CTRL-Wコマンドをトラック0に対して行うことはできません．

CTRL-W (ライトディスク)

CTRL-Rコマンドと逆の操作を行うコマンドで，メモリの内容をフロッピーディスクへセーブします．

CTRL-W〈ドライブ＃〉〔,〈サーフェス＃〉〕,〈トラック＃〉,〈セクタ＃〉,〈開始アドレス〉,〈終了アドレス〉

〈開始アドレス〉と〈終了アドレス〉は，セーブするメモリ領域を指定します．このとき，指定する領域は1セクタ単位(256バイト)である必要はありません．ただし，フロッピーディスクはセクタ単位で指定しますから，最低1セクタを使用します．

- 8インチフロッピーディスクでは，CTRL-D, CTRL-R, CTRL-Wコマンドをトラック0に対して行うことはできません．

注意　このコマンドは，ディスクへ直接書き込みを行います．ドライブ＃，サーフェス＃，トラック＃，セクタ＃の指定を間違えると，ディスクを壊す危険性があります．
ディスクの構造をよく理解したうえで，慎重にご使用ください．

第20章

機械語プログラムを呼ぶ

機械語は，BASICに比べるととっつきが悪いうえ，CPU内のレジスタやシステムについての正しい知識が要求されます．このため，慣れないうちはすぐ暴走してしまいますが，うまくいけば，BASICでは得られない実行速度やハードウェアの操作が可能となります．N88-BASIC/N88-日本語BASICには，機械語で書かれたプログラムを呼び出す機能として，USR関数とCALL文が用意されています．ここでは，機械語プログラムを作成する際の注意と，USR関数，CALL文との引数の受け渡し方について説明します．

1 メモリの確保(CLEAR)

機械語ルーチンをモニタより作成したり，ファイルからロードする際には，そのためのメモリ領域を確保する必要があります．この機械語プログラム用のメモリ領域は，BASICの使用する領域と重複してはなりません．さもないと，BASICの動作が異常になったり，逆にBASICの動作により準備した機械語プログラムが破壊されたりします．

メモリ領域の確保は，CLEAR文によりBASICの使用するメモリの上限を設定することにより行います．BASICは，指定された上限以上の領域は利用しなくなりますので，機械語プログラム領域として用いることができます(CLEAR文参照)．

2 機械語プログラムの準備

1. 3つの作成法

確保した領域に機械語プログラムを準備する主な方法としては，次の3つがあります．

(1) モニタを用いて作成する

MONコマンドにより機械語モニタに移り，モニタの機能を利用してプログラムを書き込む(「第19章　機械語モニタ」参照)．

(2) BLOADにより読み込む

すでにBSAVEによりセーブされている機械語プログラムをロードする．

(3) POKEによりプログラムを書き込む

比較的短いプログラムであれば，DATA文により機械語プログラムをデータの型で表現しておき，それをREAD文で読み出した後，POKE文により書き込む．

2. プログラム作成上の注意

また，機械語プログラムを作成する際には次の点に注意してください．

(1) BASICに制御を戻すためには，プログラムはリターン命令(RET)を用いなければなりません．

(2) 機械語ルーチンに引き渡された引数が文字列であった場合には，引数の値は次のような3バイトから成るストリングディスクリプタと呼ばれる領域のアドレスとなります．

① ストリングディスクリプタの最初のバイトは，その文字列の長さ(0から255)を保持します．

② 2番目のバイトは，文字列の置かれている先頭アドレスの下位8ビットを保持します．

③ 最後のバイトは，文字列の置かれている先頭アドレスの上位8ビットを保持します．

機械語ルーチンでは，この3バイトのストリ

ングディスクリプタを書き換えてはなりません．また，ストリングディスクリプタによって指されるストリングの内容は変更することができますが，その長さを変更してはなりません．

(3) 機械語ルーチンからBASICに戻るときには，呼び出されたときとスタックポインタ(SP)の値が等しくなっていなければなりません．

(4) 機械語ルーチンが呼び出されたとき，スタック領域としては8レベル(16バイト)分だけ利用できるように設定されています．もし機械語ルーチン内でより大きなスタックを必要とする場合には，ルーチン内で独自にスタックを用意しなければなりません．またこのとき，BASICに戻る際にスタックポインタをもとに戻すことを忘れてはなりません．

3 USR関数の呼び出し

1. USR関数の書式

USR関数は

USR〔〈番号〉〕(〈引数〉)

という書式によって呼び出されます．ただし，〈番号〉は0から9までの数値で，引数は任意の数値式または文字式です．USR関数は書式上〈引数〉を必要としますので，呼び出されたルーチンが引数を必要としない場合でも，ダミーの引数を与える必要があります．

2. 各レジスタの内容

USR関数が呼び出され，対応する機械語ルーチンに制御が移ったとき，Aレジスタは，渡された引数の型を示す値を持っています．その値と意味は次のようになっています．

Aの値	引数の型
2	整数型
3	文字列型
4	単精度型
8	倍精度型

引数が文字列の場合には，〔DE〕レジスタペアが3バイトから成るストリングディスクリプタを指しています．

引数が数値の場合には，〔HL〕レジスタに浮動小数点アキュムレータ(FAC)と呼ばれる8バイトの領域の5バイト目のアドレスが入っています．実際の引数は，このFACの内に各型に応じて以下のように格納されて渡されます．

(1) 整数型のとき

FACの5バイト目(〔HL〕)レジスタペアが指しているところ)に引数の下位8ビット，6バイト目に上位8ビットが入ります．

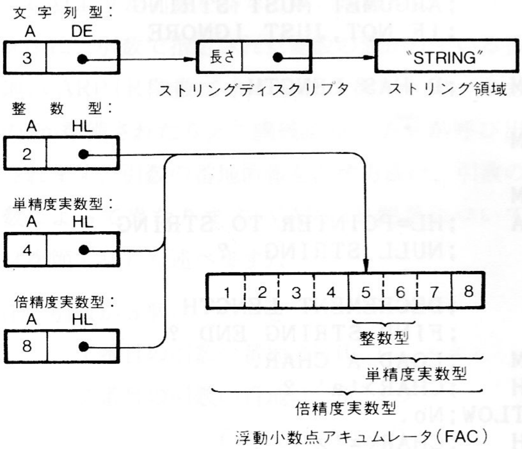
(2) 単精度実数型のとき

FACの8バイト目は指数部になっており，

(指数−128)の値が入ります．小数点は仮数部の最上位ビットの左にあると想定します．7バイト目は仮数部の最高部7ビットを保持します．このバイトの最上位ビットは符号を示し，0で正，1で負を表します．6バイト目と5バイト目は，それぞれ仮数部の中部と最低部の8ビットを保持します．

(3) 倍精度実数型のとき

5バイト目から8バイト目までは，単精度実数型と同じように指数部と仮数部の上位3バイトが入ります．1バイト目から4バイト目には，仮数部の下位4バイトが入ります．



3. BASICに値を返す

機械語プログラム中，RET命令に出会うとBASICに制御が移り，USR関数を呼び出した次の行のBASICプログラムが実行されます．

USR関数がBASICに値を返すには，FACで求めた関数値を設定してやることで行えます．通常，返す値は，引数として渡した値と同じ型になります．

4. サンプルプログラム

USR関数の例として，文字列中に含まれる小文字の数を返す関数を示します．ここで紹介するサンプルプログラムは，DATA文により機械語プログラムをあらかじめ用意し，次にPOKE文により機械語プログラムを書き込みます．これにより，機械語プログラムを実行します．

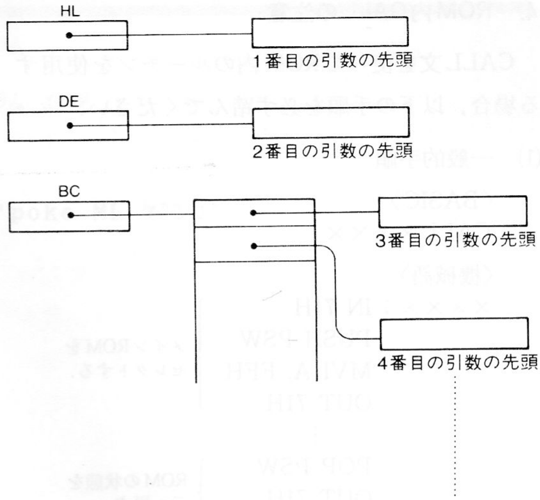
機械語ルーチンでは，まずAレジスタの値を調べて，与えられた引数が文字列かどうかを調べています．ここでは，文字列でない場合はそのままBASICに戻ることにしています．

引数が文字型であった場合には，〔DE〕レジスタにより渡されたストリングディスクリプタの番地をもとに，文字列の長さと，実際に文字列の置かれている番地を求めます．

その後，ストリングを最初から1文字ずつ調べていき，小文字であった場合には，ANI命令によりD_5ビットを0にすることにより大文字に変換し，ストリングを変更します．このようにユーザ関数は与えられた引数自身を変更するために，USR0の呼び出し前後ではA$の内容が変化することになります．

```
100 '  USR ﾌｧﾝｸｼｮﾝ ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
110 '
120 CLEAR ,&HE000  'ﾘｻﾞｰﾌﾞ ﾌﾟﾛｸﾞﾗﾑ ｴﾘｱ
130 MACHINE=&HE000
140 DEF USR0=MACHINE
150 '
160 FOR ADR=MACHINE TO MACHINE+30
170   READ BYTE:POKE ADR,BYTE   'poke ML ﾌﾟﾛｸﾞﾗﾑ
180 NEXT ADR
190 '
200 INPUT A$
210   B$=USR0(A$)
220   PRINT B$
230 GOTO 200
240 END
250 '
260 '  ﾏｼﾝｺﾞ ﾌﾟﾛｸﾞﾗﾑ ﾃﾞｰﾀ
270 '
280 DATA &HFE,&H03       :'CPI  3      ;ARGUMET MUST STRING
290 DATA &HC0            :'RNZ         ;IF NOT,JUST IGNORE
300 DATA &HEB            :'XHCG
310 DATA &H56            :'MOV  D,M    ;D HAS LENGTH
320 DATA &H23            :'INX  H
330 DATA &H7E            :'MOV  A,M
340 DATA &H23            :'INX  H
350 DATA &H66            :'MOV  H,M
360 DATA &H6F            :'MOV  L,A    ;HL=POINTER TO STRING
370 DATA &H14            :'INR  D      ;NULL STRING  ?
380 '              LOOP:
390 DATA &H15            :'DCR  D      ;DECREMENT LENGTH
400 DATA &HC8            :'RZ          ;FIND STRING END ?
410 DATA &H7E            :'MOV  A,M    ;LOAD A CHAR.
420 DATA &HFE,&H61       :'CPI  61H    ;CHAR>'a' ?
430 DATA &HDA,&H1B,&HE0:'JC   NOTLOW;No.
440 DATA &HFE,&H7B       :'CPI  7BH    ;CHAR<='z'  ?
450 DATA &HD2,&H1B,&HE0:'JNC  NOTLOW;No.
460 DATA &HE6,&HDF       :'ANI  0DFH   ;CONVERT TO UPPER
470 DATA &H77            :'MOV  M,A    ;SET IT
480 '            NOTLOW:
490 DATA &H23            :'INX  H
500 DATA &HC2,&H0B,&HE0:'JMP  LOOP
```

4 CALL文



1. 引数を持たないCALL文

機械語ルーチンはCALL文を用いても呼び出すことができます．

引数を持たないCALL文は機械語のCALL命令と同等であり，指定された番地からのルーチンが呼び出されます．BASICに戻るには，USR関数と同じようにRET命令で戻ります．

2. 引数を伴ったCALL文

CALL文が引数を持っている場合には，レジスタ中に，引数で指定された変数の置かれている番地(VARPTR関数で求められる番地と同一のもの)が準備されたうえで機械語ルーチンが呼び出されます．引数の番地情報を渡す方法は，引数の数によって異なります(VARPTR関数については次節で詳しく述べます)．

(1) 引数が3個以下のとき

1番目の引数の番地は〔HL〕レジスタ
2番目の引数の番地は〔DE〕レジスタ
(もしあるならば)
3番目の引数の番地は〔BC〕レジスタ
(もしあるならば)

に準備されます．

(2) 引数が4個以上のとき

1番目の引数の番地は〔HL〕レジスタ
2番目の引数の番地は〔DE〕レジスタ

に準備されます．

3番目以降の引数については，それらの番地が順番に並べられた表が作成され，〔BC〕レジスタにその表の番地が準備されます．

3. CALL文を用いる際には，次の事項に注意してください．

(1) CALL文が渡す引数の番地は，USR関数の場合のFACの番地ではなく，VARPTR関数の値と同じ，変数値の置かれている番地です．

(2) CALL文では，複数の引数を与えることができます．また，与えられた引数の番地を介してBASICに値を返すことができます．

(3) 引数が文字型の場合には，準備される番地はストリングディスクリプタの番地です．

(4) CALL文は，引数の個数については何ら情報を与えられません．したがって，これらを一致させることに注意しなければなりません．

4. ROM内CALLの注意

CALL文を使ってROM内のルーチンを使用する場合，以下の手順を必ず踏んでください．

(1) 一般的手順

〈BASIC〉

```
CALL ××××
```

〈機械語〉

```
××××：IN 71H
      PUSH PSW     ｝メインROMを
      MVI A, FFH   ｝セレクトする．
      OUT 71H
        ⋮
      POP PSW      ｝ROMの状態を
      OUT 71H      ｝元へ戻す．
      RET
```

(2) "A"という文字の画面出力

〈BASIC〉

```
100 CLEAR ,&HCEFF
200 ADR=&HCF00
300 CALL ADR
400 END
```

〈機械語〉

```
CF00： IN    71H
       PUSH  PSW
       MVI   A, FF
       OUT   71H
       MVI   A, 41H   ｝"A"を表示
       RST   3
       POP   PSW
       OUT   71H
       RET
```

5. サンプルプログラム

CALL文の例として，2つの数の最大値を与えるルーチンを示します．A%, B%により渡された2つの数のうちの大きい方の値をC%に代入します．

機械語ルーチンでは〔HL〕レジスタの持つ番地からA%，〔DE〕レジスタの持つ番地からB%の値を得，大きい方を〔BC〕レジスタの示すC%の番地に格納します．

```
100 '  CALL ｽﾃｰﾄﾒﾝﾄ ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
110 '
120 CLEAR,&HEØØØ   'ﾘｻﾞｰﾌﾞ ﾌﾟﾛｸﾞﾗﾑ ｴﾘｱ
13Ø MAX=&HEØØØ
14Ø '
15Ø FOR ADR=MAX TO MAX+45
16Ø   READ BYTE:POKE ADR,BYTE   'poke ML ｹﾞﾝｺﾞ
17Ø NEXT ADR
18Ø '
19Ø C%=Ø
2ØØ INPUT A%,B%
21Ø   CALL MAX(A%,B%,C%)
22Ø   PRINT C%
23Ø GOTO 2ØØ
24Ø END
25Ø '
26Ø '  ﾏｼﾝｺﾞ ﾌﾟﾛｸﾞﾗﾑ ﾃﾞｰﾀ
27Ø '
28Ø '                    MAX:
29Ø DATA &HDB,&H71       :'IN   71H
3ØØ DATA &HF5            :'PUSH PSW
31Ø DATA &H3E,&HFF       :'MVI  A,ØFFH
32Ø DATA &HD3,&H71       :'OUT  71H        ;SELECT N88 MAIN ROM
33Ø DATA &HCD,&H29,&HEØ:'CALL GETVAL     ;GET A% VALUE
34Ø DATA &HEB            :'XCHG
35Ø DATA &HCD,&H29,&HEØ:'CALL GETVAL     ;GET B% VALUE
36Ø DATA &HE5            :'PUSH H
37Ø DATA &HB7            :'ORA  A          ;CLEAR CARRY
38Ø DATA &H7B            :'MOV  A,E        ;A%-B%
39Ø DATA &H95            :'SUB  L
4ØØ DATA &H7A            :'MOV  A,D
41Ø DATA &H9C            :'SBB  H
42Ø DATA &H2E,&HFF       :'MVI  L,ØFFH
43Ø DATA &HE2,&H1A,&HEØ:'JPO  *+4        ;CHECK V FLAG
44Ø DATA &H2C            :'INR  L
45Ø DATA &HAD            :'XRA  L
46Ø DATA &HE1            :'POP  H          ;ASSUME B%<A%
47Ø DATA &HF2,&H2Ø,&HEØ:'JP   *+4
48Ø DATA &HEB            :'XCHG            ;A%<=B%
49Ø DATA &H7D            :'MOV  A,L        ;STORE INTO C%
5ØØ DATA &HØ2            :'STAX B
51Ø DATA &HØ3            :'INX  B
52Ø DATA &H7C            :'MOV  A,H
53Ø DATA &HØ2            :'STAX B
54Ø DATA &HF1            :'POP  PSW
55Ø DATA &HD3,&H71       :'OUT  71H
56Ø DATA &HC9            :'RET
57Ø '               GETVAL:
58Ø DATA &H7E            :'MOV  A,M
59Ø DATA &H23            :'INX  H
6ØØ DATA &H66            :'MOV  H,M
61Ø DATA &H6F            :'MOV  L,A
62Ø DATA &HC9            :'RET
```

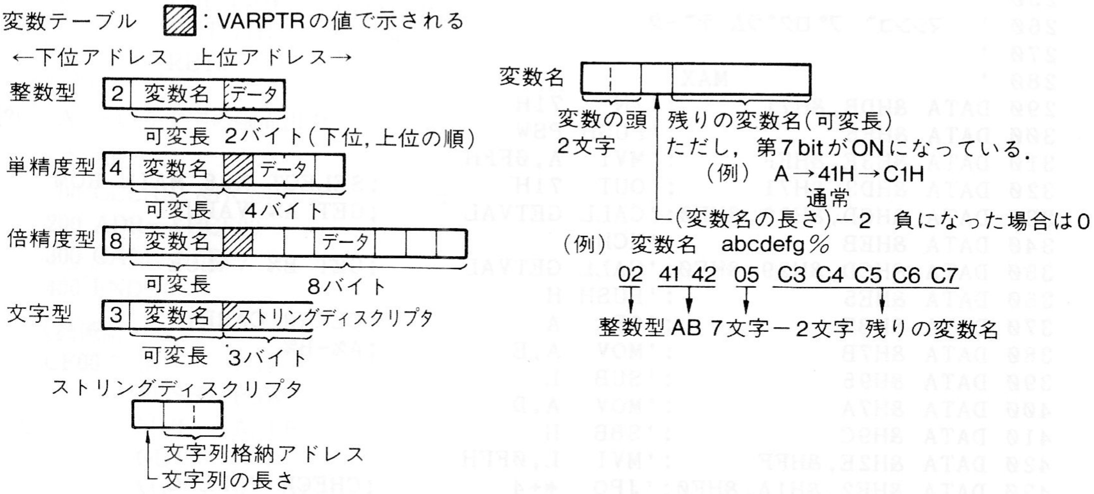
5 VARPTR

VARPTR関数は，指定された変数，バッファの置かれている番地を返します．

VARPTR関数の引数として変数名が与えられたときには，その変数の値の最下位バイトが置かれている番地を返します．上位バイトは，下図に示すようにこの番地以外に入っています．また，文字変数名が指定された場合には，ストリングディスクリプタの先頭番地が返されます．

変数として配列要素が指定された場合には，その要素の置かれている番地が返されます．

引数としてバッファ番号が与えられた場合には，そのバッファ領域の番地を返します．この領域の最初の9バイトはBASIC作業用の領域となっており，ファイルとの入出力に用いられる256バイトの領域は，VARPTR関数の値に9を加えた番地から始まります．



資料

資料1．コントロールコード

16進	10進	キー入力	動　　作
01H	1	【コントロール】＋【A ち】	【説明】と同じ
02H	2	【コントロール】＋【B こ】	1つ前のワードに戻る
03H	3	【コントロール】＋【C そ】	実行の中断（【停止】と同じ）
04H	4	【コントロール】＋【D し】	カーソル位置から1ワードを削除
05H	5	【コントロール】＋【E い】	カーソルの位置から，その行の終わりまでを消去
06H	6	【コントロール】＋【F は】	1つ先のワードへ進む
07H	7	【コントロール】＋【G き】	ブザーを鳴らす
08H	8	【コントロール】＋【H く】	1文字を削除（【削除】と同じ）
09H	9	【コントロール】＋【I に】	タブ位置までカーソルを移動（タブ位置は8桁に設定されている）（【タブ】と同じ）
0AH	10	【コントロール】＋【J ま】	ラインフィード，インサートモードで2行に分割
0BH	11	【コントロール】＋【K の】	カーソルをホームポジション（左上スミ）に移動（【シフト】＋【画面消去】と同じ）
0CH	12	【コントロール】＋【L り】	テキスト画面を消去する（【画面消去】と同じ）
0DH	13	【コントロール】＋【M も】	キャリッジリターン（【↵】と同じ）
0FH	15	【コントロール】＋【O ら】	プログラム実行中に画面の表示を無効にする
12H	18	【コントロール】＋【R す】	インサートモードにする（【挿入】と同じ）
13H	19	【コントロール】＋【S と】	実行を一時停止する
15H	21	【コントロール】＋【U な】	1行キャンセル
18H	24	【コントロール】＋【X さ】	カーソルを行の最後に移す
1CH	28	【→】	カーソルを右へ移動
1DH	29	【←】	カーソルを左へ移動
1EH	30	【↑】	カーソルを上へ移動
1FH	31	【↓】	カーソルを下へ移動

• 画面表示のときの動作

PRINT 文で CHR$ 関数を用いて出力します.

例 print chr$(7);

16進	10進	動作
07H	7	ブザーを鳴らします.
09H	9	タブ設定までカーソルを移動させます. タブ位置は8桁ごとに設定されています.
0AH	10	カーソルを1つ下の行へ移動させます.
0BH	11	カーソルをホームポジション(左上スミ)に移動させます.
0CH	12	テキスト画面を消去します.
0DH	13	カーソルを行の左端へ移動させます.
16H	22	半角文字入力モードにします.*
17H	23	全角文字入力モードにします.*
1CH	28	カーソルを1つ右へ移動させます.
1DH	29	カーソルを1つ左へ移動させます.
1EH	30	カーソルを上の行へ移動させます.
1FH	31	カーソルを下の行へ移動させます.

＊ N_{88}-日本語BASICでのみ有効です.

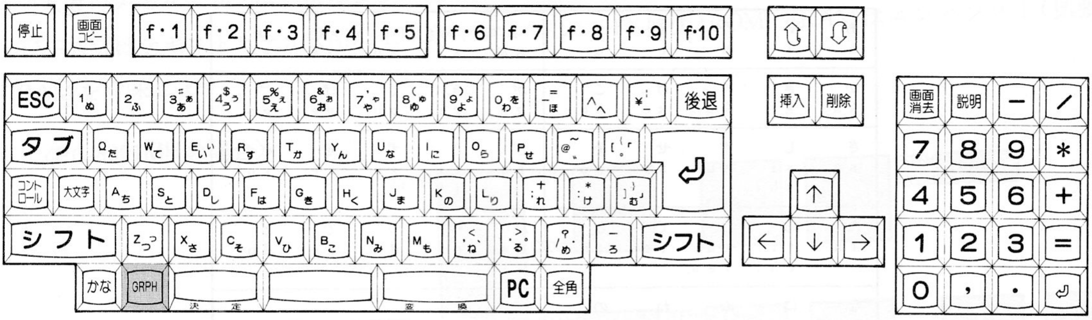
資料2. キャラクタコード

上位4ビット →

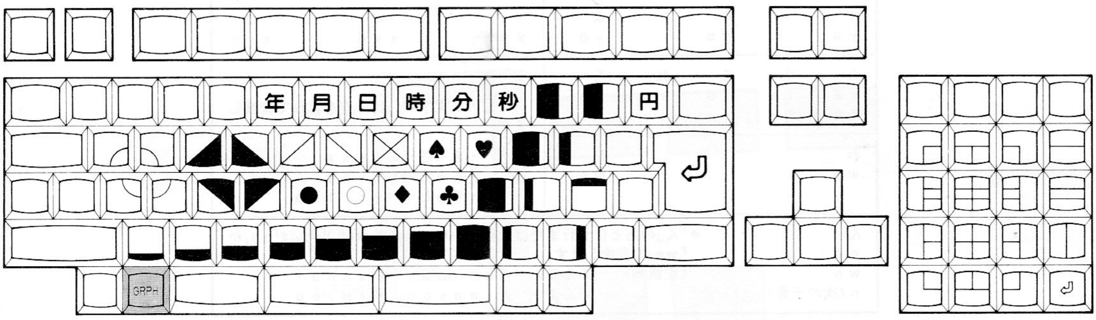
下位4ビット ↓

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		D_E		0	@	P		p				ー	タ	ミ		
1	S_H	D_1	!	1	A	Q	a	q			。	ア	チ	ム		円
2	S_X	D_2	"	2	B	R	b	r			「	イ	ツ	メ		年
3	E_X	D_3	#	3	C	S	c	s			」	ウ	テ	モ		月
4	E_T	D_4	$	4	D	T	d	t			、	エ	ト	ヤ		日
5	E_Q	N_K	%	5	E	U	e	u			・	オ	ナ	ユ		時
6	A_K	S_N	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		分
7	B_L	E_B	'	7	G	W	g	w			ァ	キ	ヌ	ラ		秒
8	B_S	C_N	(	8	H	X	h	x			ィ	ク	ネ	リ	♠	
9	H_T	E_M	)	9	I	Y	i	y			ゥ	ケ	ノ	ル	♥	
A	L_F	S_B	*	:	J	Z	j	z			ェ	コ	ハ	レ	♦	
B	H_M	E_C	+	;	K	[	k	{			ォ	サ	ヒ	ロ	♣	
C	C_L	→	,	<	L	¥	l	¦			ャ	シ	フ	ワ	●	
D	C_R	←	−	=	M	]	m	}			ュ	ス	ヘ	ン	○	
E	S_O	↑	.	>	N	^	n	~			ョ	セ	ホ	゛		
F	S_I	↓	/	?	O	_	o				ッ	ソ	マ	゜		

資料3. キーボードレイアウト



キーの配列



グラフィックシンボルキーの配列

資料4. ローマ字→かな変換規則

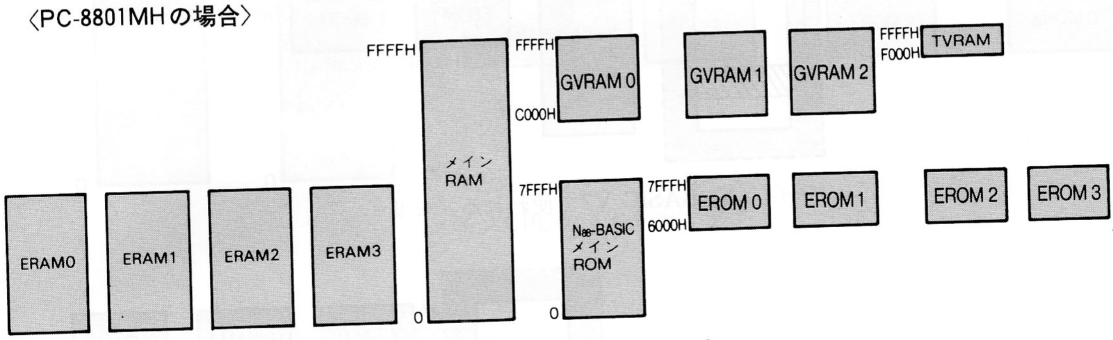
あ　い　う　え　お a　i　u　e　o	ぁ　ぃ　ぅ　ぇ　ぉ xa　xi　xu　xe　xo
か　き　く　け　こ ka　ki　ku　ke　ko	が　ぎ　ぐ　げ　ご ga　gi　gu　ge　go
さ　し　す　せ　そ sa　si　su　se　so 　shi	ざ　じ　ず　ぜ　ぞ za　zi　zu　ze　zo 　ji
た　ち　つ　て　と ta　ti　tu　te　to 　chi　tsu	だ　ぢ　づ　で　ど da　di　du　de　do
な　に　ぬ　ね　の na　ni　nu　ne　no	
は　ひ　ふ　へ　ほ ha　hi　hu　he　ho 　fu	ば　び　ぶ　べ　ぼ ba　bi　bu　be　bo ぱ　ぴ　ぷ　ぺ　ぽ pa　pi　pu　pe　po
ま　み　む　め　も ma　mi　mu　me　mo	
や　ゆ　よ ya　yu　yo	ゃ　ゅ　ょ xya　xyu　xyo
ら　り　る　れ　ろ ra　ri　ru　re　ro la　li　lu　le　lo	っ かなの小文字「っ」は子音を重ねて変換します。 〈例〉・きっさ→kissa ・ちっそ→tisso ・りっとう→rittou
わ　を wa　wo	
ん nn wn n (次の子音)	●「ん」のあとにナ行またはヤ行のかながくるときは、「ん」は「nn」か「wn」で変換します。 〈正誤例〉・あんない　正annnai　誤annai ・いんよう　正innyou　誤inyou

きゃ　きゅ　きょ kya　kyu　kyo	ぎゃ　ぎゅ　ぎょ gya　gyu　gyo
しゃ　しゅ　しぇ　しょ sha　shu　she　sho sya　syu　sye　syo	じゃ　じゅ　じぇ　じょ ja　ju　je　jo zya　zyu　zye　zyo
ちゃ　ちゅ　ちぇ　ちょ cha　chu　che　cho tya　tyu　tye　tyo	ぢゃ　ぢゅ　ぢょ dya　dyu　dyo
にゃ　にゅ　にょ nya　nyu　nyo	
ひゃ　ひゅ　ひょ hya　hyu　hyo	びゃ　びゅ　びょ bya　byu　byo ぴゃ　ぴゅ　ぴょ pya　pyu　pyo
ふぁ　ふぃ　ふぇ　ふぉ fa　fi　fe　fo ふゃ　ふゅ　ふょ fya　fyu　fyo	ヴァ　ヴィ　ヴゥ　ヴェ　ヴォ va　vi　vu　ve　vo ●ヴァ行はカタカナとして入力され、これらをひらがなに変換することはできません。
みゃ　みゅ　みょ mya　myu　myo	●文字以外の変換 ー(長音)　←　－(マイナス記号) 。(句点)　←　.(ピリオド) 、(読点)　←　,(カンマ) テンキーから入力した場合は、マイナス、ピリオド、カンマのまま入力されます。
りゃ　りゅ　りょ rya　ryu　ryo lya　lyu　lyo	

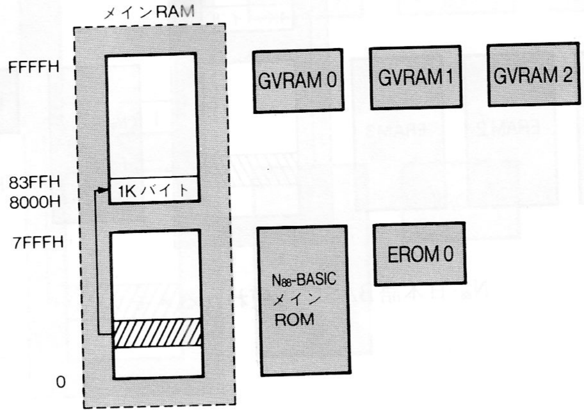
資料5．メモリマップ

本機のメモリマップは，次のようになっています．ここで，GVRAMはグラフィックビデオRAM，TVRAMはテキストビデオRAM，EROMはエンハンスト（拡張）ROM，ERAMはエンハンスト（拡張）RAMを意味します．

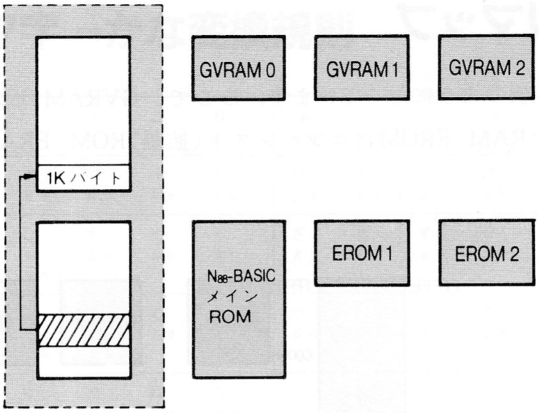
〈PC-8801MHの場合〉



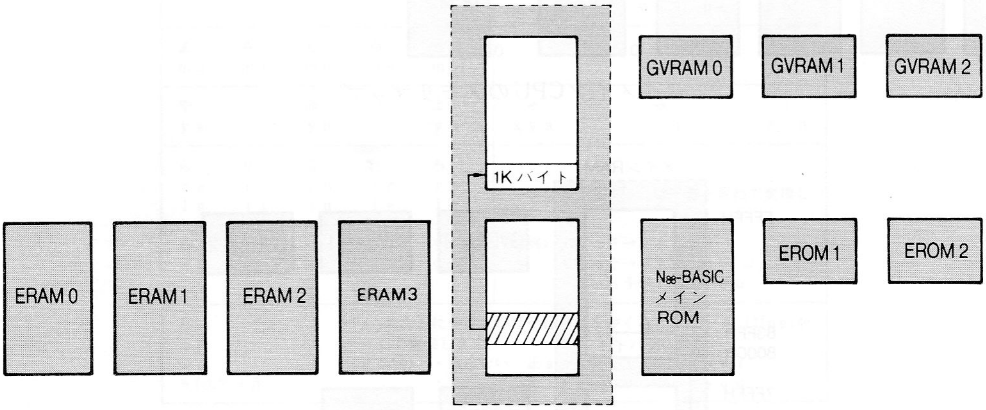
メインCPUのメモリマップ



N_{88}-BASIC V1で使われるメモリ

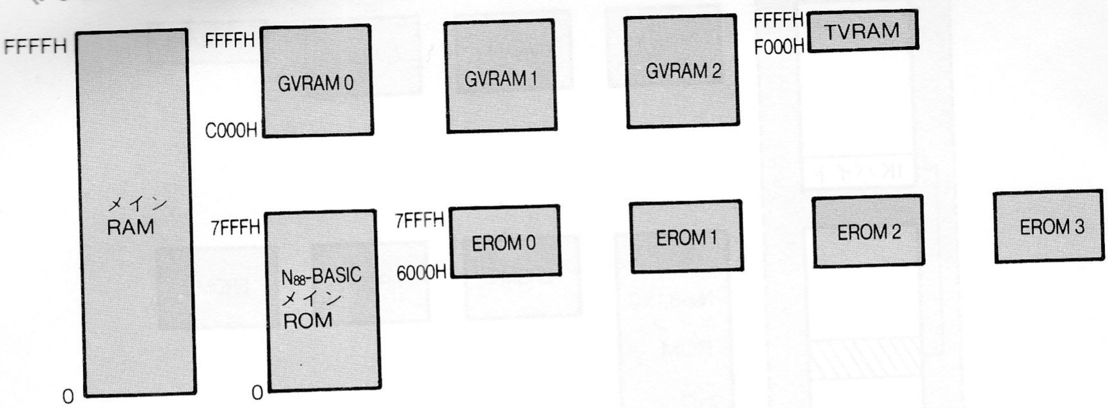


N_{88}-BASIC V2で使われるメモリ

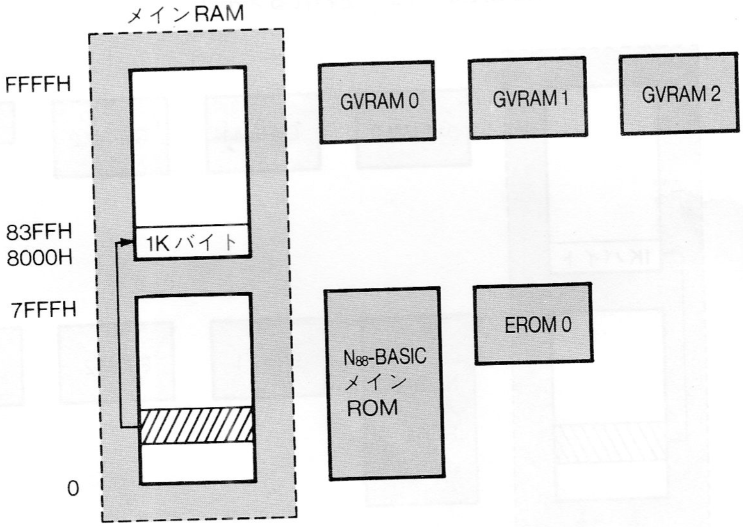


N_{88}-日本語BASICで使われるメモリ

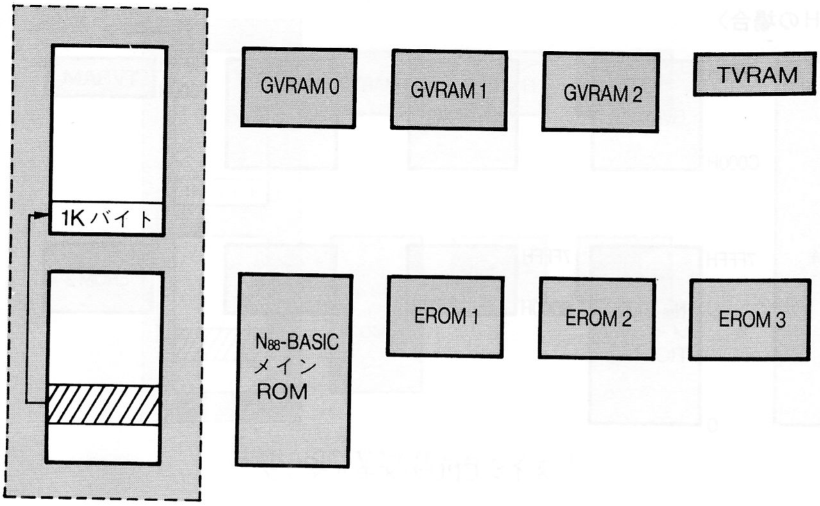
〈PC-8801FHの場合〉



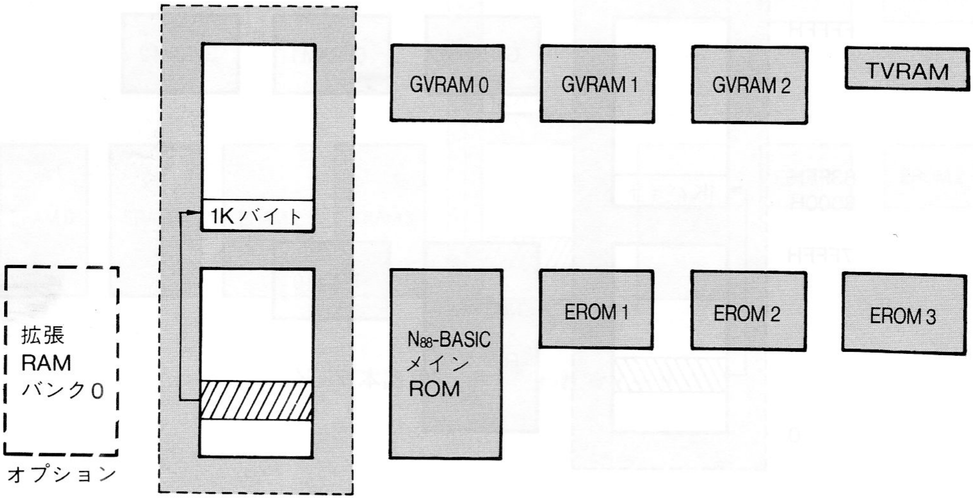
メインCPUのメモリマップ



N_{88}-BASIC V1で使われるメモリ



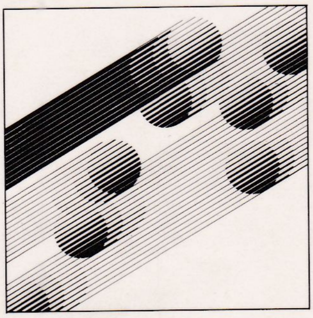
N88-BASIC V2で使われるメモリ



N88-日本語BASICで使われるメモリ

N88-日本語BASICの使用時は，拡張RAMがない場合，プログラムテキストエリアは16Kバイトになります(N88-BASICでは32Kバイトです).

拡張RAMを装着することによって，プログラムテキストエリアとして32K使用できます.



NEC 日本電気株式会社・日本電気ホームエレクトロニクス株式会社

78118612